U.S. Government Protection Profile for Single-level Operating Systems in Environments Requiring Basic Robustness

Version 0.3

Information Assurance Directorate

National Security Agency
9800 Savage Road
Fort George G. Meade, MD 20755-6000

29 January 2004

Foreword

1 This publication, "*U.S. Government Protection Profile for Single-level Operating Systems in Environments Requiring Basic Robustness",* is issued by the Information Assurance Directorate as part of its program to promulgate security standards for information systems. This protection profile is based on the "Common Criteria for Information Technology Security Evaluations, Version 2.1."

2 Further information, including the status and updates, of this protection profile can be found on the internet at: http://www.iatf.net/protection_profiles/index.cfm.

3 Comments on this document should be directed to: ppcomments@iatf.net. The comments should include the title of the document, the page, the section number, and paragraph number, detailed comment and recommendations.

Table of Contents

List of Figures

List of Tables

1. Introduction

4 This section contains overview information necessary to allow a Protection Profile (PP) to be registered through a Protection Profile Registry. The PP identification provides the labeling and descriptive information necessary to identify, catalogue, register, and cross-reference a PP. The PP overview summarizes the profile in narrative form and provides sufficient information for a potential user to determine whether the PP is of interest. The overview can also be used as a stand-alone abstract for PP catalogues and registers. The "Conventions" section provides the notation, formatting, and conventions used in this protection profile. The "Glossary of Terms" section gives a basic definition of terms, which are specific to this PP. The "Document Organization" section briefly explains how this document is organized.

1.1 Identification

5 Title: U.S. Government Protection Profile for Single-level Operating Systems in Environments Requiring Basic Robustness Version 0.3, 29 January 2004

6 Registration: < to be provided upon registration >

7 Keywords: operating system, COTS, commercial security, basic robustness, access control, discretionary access control, DAC

1.2 Overview

8 The "*Protection Profile for Operating Systems Implementing Commercial Security"* specifies security requirements for commercial-off-the-shelf (COTS) general-purpose operating systems in networked environments. This profile establishes the requirements necessary to achieve the security objectives of the Target of Evaluation (TOE) and its environment.

9 Conformant products support Identification and Authentication, Discretionary Access Control (DAC), and an audit capability. These systems provide adequate security services, mechanisms, and assurances to process administrative, private, and sensitive/proprietary information. When an organization's most sensitive/proprietary information is to be sent over a publicly accessed network, the organization should apply additional protection at the network boundaries.

1.2.1 TOE Environment Defining Factors

10 In trying to specify the environments in which TOEs with various levels of robustness are appropriate, it is useful to first discuss the two defining factors that characterize that environment: **value of the resources** and **authorization of the entities** to those resources.

11 In general terms, the environment for a TOE can be characterized by the authorization (or lack of authorization) the least trustworthy entity has with respect to the highest value of TOE resources (i.e. the TOE itself and all of the data processed by the TOE).

12 Note that there are an infinite number of combinations of entity authorization and value of resources; this conceptually "makes sense" because there are an infinite number of potential environments, depending on how the resources are valued by the organization, and the variety of authorizations the organization defines for the associated entities. In the next section 1.2.2, these two environmental factors will be related to the robustness required for selection of an appropriate TOE.

1.2.1.1 Value of Resources

13 Value of the resources associated with the TOE includes the data being processed or used by the TOE, as well as the TOE itself (for example, a real-time control processor). "Value" is assigned by the using organization. For example, in the DoD low-value data might be equivalent to data marked "FOUO", while high-value data may be those classified Top Secret. In a commercial enterprise, low-value data might be the internal organizational structure as captured in the corporate on-line phone book, while high-value data might be corporate research results for the next generation product. Note that when considering the value of the data one must also consider the value of data or resources that are accessible through exploitation of the TOE. For example, a firewall may have "low value" data itself, but it might protect an enclave with high value data. If the firewall was being depended upon to protect the high value data, then it must be treated as a high-value-data TOE.

1.2.1.2 Authorization of Entities

14 Authorization that entities (users, administrators, other IT systems) have with respect to the TOE (and thus the resources of that TOE, including the TOE itself) is an abstract concept reflecting a combination of the trustworthiness of an entity and the access and privileges granted to that entity with respect to the resources of the TOE. For instance, entities that have total authorization to all data on the TOE are at one end of this spectrum; these entities may have privileges that allow them to read, write, and modify anything on the TOE, including all TSF data. Entities at the other end of the spectrum are those that are authorized to few or no TOE resources. For example, in the case of a router, non-administrative entities may have their packets routed by the TOE, but that is the extent of their authorization to the TOE's resources. In the case of an OS, an entity may not be allowed to log on to the TOE at all (that is, they are not valid users listed in the OS's user database)**.**

15 It is important to note that authorization **does not** refer to the **access** that the entities actually have to the TOE or its data. For example, suppose the owner of the system determines that no one other than employees was authorized to certain data on a TOE, yet they connect the TOE to the Internet. There are millions of entities that are not **authorized** to the data (because they are not employees), but they actually have connectivity to the TOE through the Internet and thus can attempt to access the TOE and its associated resources.

16 Entities are characterized according to the value of resources to which they are authorized; the extent of their authorization is implicitly a measure of how trustworthy the entity is with respect to compromise of the data (that is, compromise of any of the applicable security policies; e.g.,

confidentiality, integrity, availability). In other words, in this model the greater the extent of an entity's authorization, the more trustworthy (with respect to applicable policies) that entity is.

1.2.2 Selection of Appropriate Robustness Levels

17 Robustness is a characteristic of a TOE defining how well it can protect itself and its resources; a more robust TOE is better able to protect itself. This section relates the defining factors of IT environments, authorization, and value of resources to the selection of appropriate robustness levels.

18 When assessing any environment with respect to Information Assurance the critical point to consider is the likelihood of an attempted security policy compromise, which was characterized in the previous section in terms of entity authorization and resource value. As previously mentioned, robustness is a characteristic of a TOE that reflects the extent to which a TOE can protect itself and its resources. It follows that as the likelihood of an attempted resource compromise increases, the robustness of an appropriate TOE should also increase.

19 It is critical to note that several combinations of the environmental factors will result in environments in which the likelihood of an attempted security policy compromise is similar. Consider the following two cases:

20 The first case is a TOE that processes only low-value data. Although the organization has stated that only its employees are authorized to log on to the system and access the data, the system is connected to the Internet to allow authorized employees to access the system from home. In this case, the least trusted entities would be unauthorized entities (e.g. non-employees) exposed to the TOE because of the Internet connectivity. However, since only low-value data are being processed, the likelihood that unauthorized entities would find it worth their while to attempt to compromise the data on the system is low and selection of a basic robustness TOE would be appropriate.

21 The second case is a TOE that processes high-value (e.g., classified) information. The organization requires that the TOE be stand-alone, and that every user with physical and logical access to the TOE undergo an investigation so that they are authorized to the highest value data on the TOE. Because of the extensive checks done during this investigation, the organization is assured that only highly trusted users are authorized to use the TOE. In this case, even though high value information is being processed, it is unlikely that a compromise of that data will be attempted because of the authorization and trustworthiness of the users and once again, selection of a basic robustness TOE would be appropriate.

22 The preceding examples demonstrated that it is possible for radically different combinations of entity authorization/resource values to result in a similar likelihood of an attempted compromise. As mentioned earlier, the robustness of a system is an indication of the protection being provided to counter compromise attempts. Therefore, a basic robustness system should be sufficient to counter compromise attempts where the likelihood of an attempted compromise is low. The following chart depicts the “universe” of environments characterized by the two factors discussed in the previous section: on one axis is the authorization defined for the least

trustworthy entity, and on the other axis is the highest value of resources associated with the TOE.

23 As depicted in the following figure, the robustness of the TOE's required in each environment steadily increases as one goes from the upper left of the chart to the lower right; this corresponds to the need to counter increasingly likely attack attempts by the least trustworthy entities in the environment. Note that the shading of the chart is intended to reflects the notion that different environments engender similar levels of "likelihood of attempted compromise", signified by a similar color. Further, the delineations between such environments are not stark, but rather are finely grained and gradual.

24 While it would be possible to create many different "levels of robustness" at small intervals along the "Increasing Robustness Requirements" line to counter the increasing likelihood of attempted compromise due to those attacks, it would not be practical or particularly useful. Instead, in order to implement the robustness strategy where there are only three robustness levels: Basic, Medium, and High, the graph is divided into three sections, with each section corresponding to set of environments where the likelihood of attempted compromise is roughly similar. This is graphically depicted in the Figure 1.

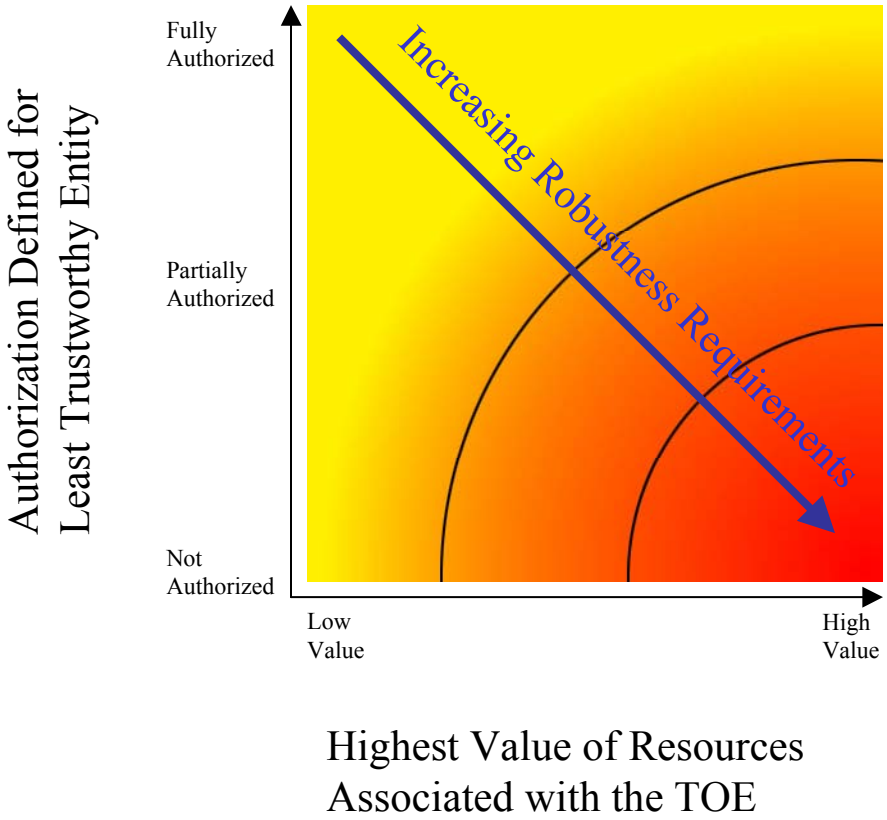

Figure 1-1 Universe of Environments

25 In this second representation of environments and the robustness plane below, Figure 2, the "dots" represent given instantiations of environments; arched lines define environments with a

similar likelihood of attempted compromise. Correspondingly, a TOE with a given robustness should provide sufficient protection for environments characterized within these arched lines. In choosing the appropriateness of a given robustness level TOE PP for an environment, then, the user must first consider the lowest authorization for an entity as well as the highest value of the resources in that environment. This should result in a "point" in the graph above, corresponding to the likelihood that that entity will attempt to compromise the most valuable resource in the environment. The appropriate robustness level for the specified TOE to counter this likelihood can then be chosen.

26 The difficult part of this activity is differentiating the authorization of various entities, as well as determining the relative values of resources; (e.g., what constitutes "low value" data vs. "medium value" data). Because every organization will be different, a rigorous definition is not possible. In <PP Section>[1] of this PP, the targeted threat level for a basic robustness TOE is characterized. This information is provided to help organizations using this PP insure that the functional requirements specified by this basic robustness PP are appropriate for their intended application of a compliant TOE.

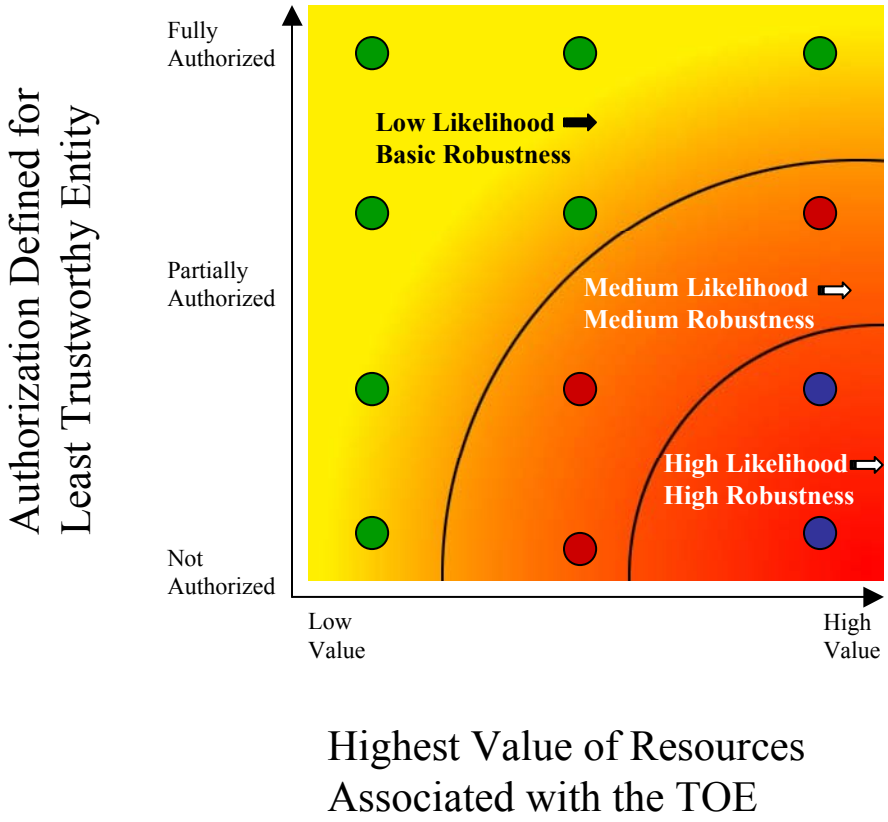

Figure 1-2 Likelihood of Attempted Compromise

[1] The PP author should insert the section of the PP that describes the TOE Environment.

1.3 Conventions

27 The notation, formatting, and conventions used in this protection profile (PP) are consistent with version 2.1 of the Common Criteria for Information Technology Security Evaluation. Font style and clarifying information conventions were developed to aid the reader.

28 The CC permits four functional component operations: assignment, iteration, refinement, and selection to be performed on functional requirements. These operations are defined in Common Criteria, Part 2, paragraph 2.1.4 as:

- Assignment: allows the specification of an identified parameter;
- Refinement: allows the addition of details or the narrowing of requirements;
- Selection: allows the specification of one or more elements from a list; and
- Iteration: allows a component to be used more than once with varying operations.

29 *Assignments or selections* left to be specified by the developer in subsequent security target documentation are italicized and identified between brackets ("[ ]"). In addition, when an assignment or selection has been left to the discretion of the developer, the text "assignment:" or "selection:" is indicated within the brackets. Assignments or selection created by the PP author (for the developer to complete) are bold, italicized, and between brackets ("[ ]"). CC selections completed by the PP author are underlined and CC assignments completed by the PP author are bold.

30 *Refinements* are identified with "**Refinement:**" right after the short name. They permit the addition of extra detail when the component is used. The underlying notion of a refinement is that of narrowing. There are two types of narrowing possible: narrowing of implementation and narrowing of scope[2]. Additions to the CC text are specified in bold. Deletions of the CC text are identified in the "End Notes" with a bold number after the element ("**8**").

31 *Iterations* are identified with a number inside parentheses ("(#)"). These follow the short family name and allow components to be used more than once with varying operations.

32 *Explicit Requirements* are allowed to create requirements should the Common Criteria not offer suitable requirements to meet the PP needs. The naming convention for explicit requirements is the same as that used in the CC. To ensure these requirements are explicitly identified, the ending "_EXP" is appended to the newly created short name.

33 *Application Notes* are used to provide the reader with additional requirement understanding or to clarify the author's intent. These are italicized and usually appear following the element needing clarification.

34 These conventions are expressed by using combinations of bolded, italicized, and underlined text as specified in Table 1.1.

[2] US interpretation #0362: Scope of Permitted Refinements

Table 1.1 - Functional Requirements Operation Conventions

Convention	Purpose	Operation
Bold	The purpose of bolded text is used to alert the reader that additional text has been added to the CC. This could be an assignment that was completed by the PP author or a refinement to the CC statement. Examples: FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records. FTA_MCS.1.1 **Refinement:** The TSF shall restrict the maximum number of concurrent **interactive** sessions that belong to the same user.	(Completed) Assignment or Refinement
Italics	The purpose of italicized text is to inform the reader of an assignment or selection operation to be completed by the developer or ST author. It has been left as it appears in the CC requirement statement. Examples: FTA_SSL.1.1The TSF shall lock an interactive session after *[assignment: a time interval of user inactivity]* by: a) Clearing or overwriting display devices, making the current contents unreadable. b) Disabling any activity of the user's data access/display devices other than unlocking the session. FDP_RIP.1.1 **Refinement:** The TSF shall ensure that any previous information content of a resource is made unavailable upon *[selection: allocation of the resource to, deallocation of the resource from]* **shared memory and operating system controlled files**.	Assignment (to be completed by developer or ST author) or Selection (to be completed by developer or ST author)
<u>Underline</u>	The purpose of underlined text is to inform the reader that a choice was made from a list provided by the CC selection operation statement. Example: FAU_STG.1.2 The TSF shall be able to <u>prevent</u> modifications to the audit records.	Selection (completed by PP author)

Convention	Purpose	Operation
Bold & Italics	The purpose of bolded and italicized text is to inform the reader that the author has added new text to the requirement and that an additional vendor action needs to be taken. Example: FIA_UAU.1.1 **Refinement:** The TSF shall allow **read access to *[assignment: list of public objects]*** on behalf of the user to be performed before the user is authenticated.	Assignment (added by the PP author for the developer or ST author to complete)
Parentheses (Iteration #)	The purpose of using parentheses and an iteration number is to inform the reader that the author has selected a new field of assignments or selections with the same requirement and that the requirement will be used multiple times. Iterations are performed at the component level. The component behavior name includes information specific to the iteration between parentheses. Example: 5.5.3.1 Management of TSF Data (for general TSF data) (FMT_MTD.1(1)) FMT_MTD.1.1(1) The TSF shall restrict the ability to **create**, query, modify, delete, and clear the **security-relevant TSF data except for audit records, user security attributes, and authentication data** to **the authorized administrator**. 5.5.3.2 Management of TSF Data (for audit records) (FMT_MTD.1(2)) FMT_MTD.1.1(2) The TSF shall restrict the ability to query, delete, and clear the **audit records** to **authorized administrators**.	Iteration 1 (of component) Iteration 2 (of component)

Convention	Purpose	Operation
Explicit: (**_EXP**)	The purpose of using **Explicit:** before the family or component behavior name is to alert the reader and to explicitly identify a newly created component. To ensure these requirements are explicitly identified, the "**_EXP**" is appended to the newly created short name and the family or component name is bolded. Example: **5.5.7.1 Explicit: Internal TSF Data Consistency (FPT_TRC_EXP.1)** **FPT_TRC_EXP.1.1** The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.	Explicit Requirement
Endnotes	The purpose of endnotes is to alert the reader that the author has deleted Common Criteria text. An endnote number is inserted at the end of the requirement, and the endnote is recorded on the last page of the section. The endnote statement first states that a deletion was performed and then provides the rationale. Following is the family behavior or requirement in its original and modified form. A strikethrough is used to identify deleted text and bold for added text. A text deletion rationale is provided. Examples: Text as shown: FDP_ACF.1.3 **Refinement:** The TSF shall explicitly authorize access of subjects to **operating system controlled files** based on the following additional rules:[15] a) Authorized administrators must follow the above-stated Discretionary Access Control policy, except after taking the following specific actions: *[assignment: list of specific actions]*. Endnote statement: [15] A deletion of CC text was performed in FDP_ACF.1.3. Rationale: The word "objects" was deleted and replaced with "operating system controlled files" to refine the scope of SFP controlled objects. FDP_ACF.1.3 **Refinement:** The TSF shall explicitly authorize access of subjects to ~~objects~~ **operating system controlled files** based on the following additional rules….	Refinement

1.4 Glossary of Terms

35 This profile uses the terms described in this section to aid in the application of the requirements. The numbers specified between brackets ("[#]") at the end of some definitions point to the "References" section to identify where these definitions were obtained.

Access	A specific type of interaction between a subject and an object that results in the flow of information from one to the other [4].
Access Control	Security service that controls the use of resources and the disclosure and modification of data
Accountability	Property that allows activities in an IT system to be traced to the entity responsible for the activity.
Administrator	A user who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TSP.
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce its security policy.
Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	Security measure that verifies a claimed identity.
Authentication data	Information used to verify a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Authorized user	An authenticated user who may, in accordance with the TSP, perform an operation.
Availability	Timely, reliable access to IT resources.
Component	The smallest selectable set of elements that may be included in a PP, an ST, or a package.
Compromise	Violation of a security policy.
Confidentiality	A security policy pertaining to disclosure of data.

Defense-in-Depth	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
Discretionary Access Control (DAC)	A means of restricting access to objects based on the identity of subjects and/or groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject [4].
Element	Individual requirements within a CC component; cannot be selected individually for inclusion in a PP, ST, or package.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or based on physical location and proximity [2].
Entity	A subject, object, user or other IT device, which interacts with TOE objects, data or resources.
Evaluation Assurance Level (EAL)	A package consisting of assurance components from CC, part 3 that represents a point on the CC predefined assurance scale.
Identity	An identifier (e.g., character string) uniquely identifying an authorized user of the TOE.
Named Object[3]	An object that exhibits all of the following characteristics: - The object may be used to transfer information between subjects of differing user identities within the TSF. - Subjects in the TOE must be able to request a specific instance of the object. - The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.
Object	An entity within the TOE security functions scope of control (TSC) that contains or receives information and upon which subjects perform operations.

[3]The only named objects in this PP, are operating system controlled files.

Operating Environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative, and personnel controls [2].
Persistent storage	All types of data storage media that maintains data across system boots (e.g., hard disk, CD, DVD).
Public Object	An object for which the TSF unconditionally permits all entities "read" access. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.
Secure State	Condition in which all TOE security policies are enforced.
Security attributes	TSF data associated with subjects, objects and users that are used for the enforcement of the TSP.
Single-level system	A system that is used to process data of a single security level.
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Sensitive information	Information that, as determined by a competent authority, must be protected because its unauthorized disclosure, alteration, loss, or destruction will at least cause perceivable damage to someone or something. [4]
Subject	An entity within the TSC that causes operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation [1].
Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
TOE Security Functions (TSF)	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP [1].

Unauthorized user	A user who may obtain access only to system provided public objects if any exist.
User	Any entity (human user or external IT entity) outside of the TOE that interacts with the TOE.
Vulnerability	A weakness that can be exploited to violate the TOE security policy.

1.5 Document Organization

36 *Section 1* provides the introductory material for the protection profile.

37 *Section 2* describes the Target of Evaluation in terms of its envisaged usage and connectivity.

38 *Section 3* defines the expected TOE security environment in terms of the threats to its security, the security assumptions made about its use, and the security policies that must be followed.

39 *Section 4* identifies the security objectives derived from these threats and policies.

40 *Section 5* identifies and defines the security functional requirements from the CC that must be met by the TOE in order for the functionality-based objectives to be met.

41 *Section 6* identifies the security assurance requirements.

42 *Section 7* provides a rationale to explicitly demonstrate that the information technology security objectives satisfy the policies and threats. Arguments are provided for the coverage of each policy and threat. The section then explains how the set of requirements are complete relative to the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective.

43 *Section 8* identifies background material used as reference to create this profile.

44 *Appendix A* defines frequently used acronyms.

2. Target of Evaluation (TOE) Description

2.1 Product Type

45 This protection profile specifies requirements for general-purpose multi-user COTS operating systems together with the underlying hardware for use in National Security Systems. Such operating systems are typically employed in a networked office automation environment (see Figure 2.1) containing file systems, printing services, network services and data archival services and can host other applications (e.g., mail, databases). This profile does not specify any security characteristics of security hardened devices (e.g. guards, firewalls) that provide environment protection at network boundaries. **When this TOE is used in composition with other systems to make up a larger system environment, the boundary protection must provide the appropriate security mechanisms and assurances as approved by NSA to ensure adequate protection for the security and integrity of this TOE and the information it protects.**

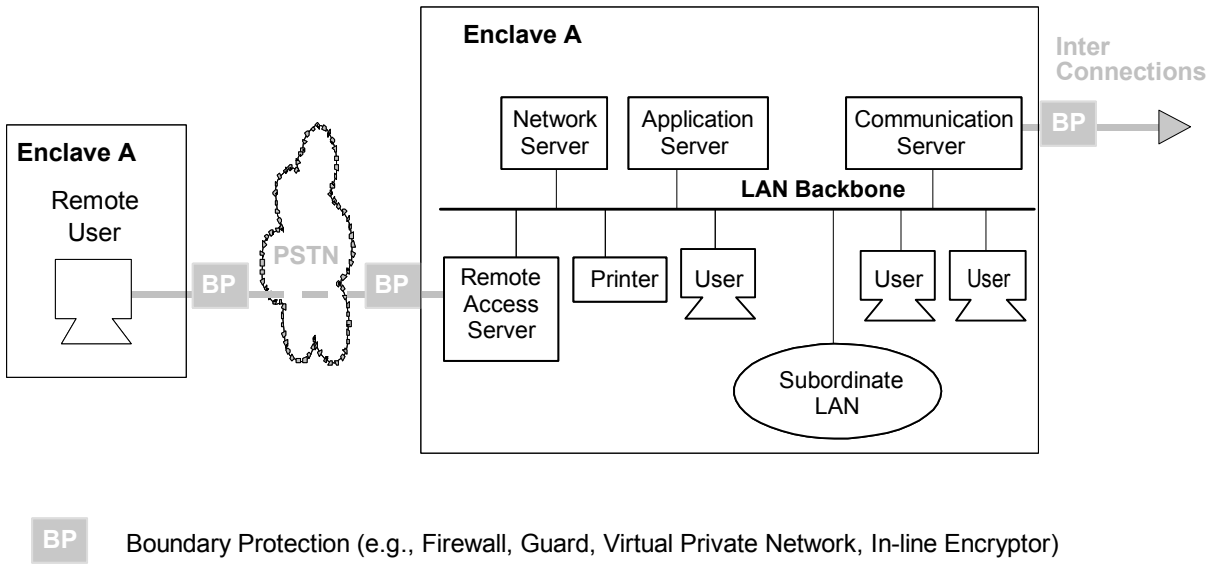

Figure 2-1 TOE Environment

2.2 General TOE Functionality

46 Conformant operating systems include the following security features:

– Identification and Authentication which mandates authorized users to be uniquely identified and authenticated before accessing information stored on the system;

- Discretionary Access Control (DAC) which restricts access to objects based on the identity of subjects and/or groups to which they belong, and allows authorized users to specify protection for objects that they control;
- Audit services which allow authorized administrators to detect and analyze potential security violations.

47 Requirements not addressed in this PP include:

48 mechanisms or services to ensure availability of data residing on the TOE. [If availability requirements exist, the environment must provide the required mechanisms (e.g., mirrored/duplicated data)],

49 mechanisms or services to ensure integrity of user data residing on the TOE, and

50 complete physical protection mechanisms, which must likewise be provided by the environment.

2.3 TOE Operational Environment

51 It is assumed that the TOE environment is under the control of a single authority and has a homogeneous security policy, including personnel and physical security. This environment can be specific to an organization or a mission and may also contain multiple networks or enclaves. They may be logical, such as an operational area network (OAN) or be based on physical location and proximity.

52 The TOE may be accessible by external IT systems that are beyond the environment's security policies. The users of these external IT systems are similarly beyond the control of the operating system's policies. Although the users of these external systems are authorized in their environments, they are outside the scope of control of this particular environment so nothing can be presumed about their intent. They must be viewed as potentially hostile.

53 This PP is appropriate for protection of administrative, private, and sensitive/proprietary information. When an organization's most sensitive information is to be sent over a publicly accessible network, the organization should consider applying additional layered security mechanisms.

3. TOE Security Environment

54 This section defines the expected TOE security environment in terms of the threats, security, assumptions, and the security policies that must be followed for the basic robustness TOE.

3.1 Use of Basic Robustness

55 Basic Robustness TOEs fall in the upper left area of the previously discussed robustness fiures. A Basic Robustness TOE is considered sufficient for low threat environments or where compromise of protected information will not have a significant impact on mission objectives. This implies that the motivation of the threat agents will be low in environments that are suitable for TOEs of this robustness. In general, basic robustness results in "good commercial practices" that counter threats based in casual and accidental disclosure or compromise of data protected by the TOE.

56 Threat agents motivation can be reconsidered in a variety of ways. One possibility is that the value of the data process are protected by the TOE will generally be seen as of little value to the adversary (i.e., compromise will have little or no impact on mission objectives). Another possibility, (where higher value data is processed or protected by the TOE) is that procuring organizations will provide other controls or safeguards (i.e., controls that the TOE itself does not enforce) in the fielded system in order to increase the threat agent motivation level for compromise beyond a level of what is considered reasonable or expected to be applied.

3.2 Threat Agent Characteristics

57 In additions to helping define the robustness appropriate for I given environment, the threat agent is a key component of the formal threat statements in the PP. Threat Agents are typically characterized by a number of factors such as motivation, expertise, and available resources. Because each robustness level is associated with a variety of environments, there are corresponding varieties of specific threat agents (that is, the threat agents will have different combinations of motivation, expertise, and available resources) that are valid for a given level of robustness. The following discussion explores the impact of each of the thereat agent factors on the ability of the TOE to protect itself (that is, the robustness required of the TOE).

58 The *motivation* of the threat agent seems to be the primary factor of the three characteristics of threat agents outlined above. Given the same expertise and set of resources, an attacker with low motivation may not be as likely to attempt to compromise the TOE. For example, an entity with no authorization to low value data none-the-less has low motivation to compromise the data; thus a basic robustness TOE should offer sufficient protection. Likewise, the fully authorized user with access to highly valued data similarly has low motivation to attempt to compromise the data, thus again a basic robustness TOE should be sufficient.

59 Unlike the motivation factor, however, the same can't be said for expertise. A threat agent with low motivation and low expertise is just as unlikely to attempt to compromise a TOE as an

attacker with low motivation and high expertise; this is because the attacker with high expertise does not have the motivation to compromise the TOE even though they may have the expertise to do so. The same argument can be made for resources as well.

60 Therefore, when assessing the robustness needed for a TOE, the motivation of threat agents should be considered a "high water mark". *That is, the robustness of the TOE should increase as the motivation of the threat agents increases.*

61 Having said that, the relationship between expertise and resources is somewhat more complicated. In general, if resources include factors other than just raw processing power (money, for example), then expertise should be considered to be at the same "level" (low, medium, high, for example) as the resources because money can be used to purchase expertise. Expertise in some ways is different, because expertise in and of itself does not automatically procure resources. However, it may be plausible that someone with high expertise can procure the requisite amount of resources by virtue of that expertise (for example, hacking into a bank to obtain money in order to obtain other resources).

62 It may not make sense to distinguish between these two factors; in general, it appears that the only effect these may have is to lower the robustness requirements. For instance, suppose an organization determines that, because of the value of the resources processed by the TOE and the trustworthiness of the entities that can access the TOE, the motivation of those entities would be "medium". This normally indicates that a medium robustness TOE would be required because the likelihood that those entities would attempt to compromise the TOE to get at those resources is the "medium" range. However, now suppose the organization determines that the entities (threat agents) that are the least trustworthy have no resources and are unsophisticated. In this case, even though those threat agents have medium motivation, the likelihood that they would be able to mount a successful attack on the TOE would be low, and so a basic robustness TOE ma be sufficient to counter that threat.

63 It should be clear from this discussion that there is no "cookbook" or mathematical answer to the question of how to specify exactly the level of motivation, the amount of resources, and the degree of expertise for a threat agent so that the robustness level of TOEs facing those threat agents can be rigorously determined. However, an organization can look at combinations of these factors applicable to their environment; discuss the issues raised in the previous paragraph; consult with appropriate accreditation authorities for input; and document their decision regarding likely threat agents in their environment.

64 The important general points we can make are:

65 The motivation for the threat agent defines the upper bound with respect to the level of robustness required for the TOE.

66 A threat agent's expertise and/or resources that is "lower" than the threat agent's motivation (e.g., a threat agent with high motivation but little expertise and few resources) may lessen the robustness requirements for the TOE (see next point, however).

67 The availability of attacks associated with high expertise and/or high availability of resources (for example, via the Internet or "hacker chat rooms") introduces a problem when trying to define the expertise of, or resources available to, a threat agent.

3.3 Threats

68 The following are the threat statements that the TOE must address.

T.ADMIN_ERROR	An administration may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An authorized administrator's intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking an user's actions.
T.POOR_DESIGN	Unintentional or intentional errors in requirement specification or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.
T.POOR_IMPLEMENTATION	Unintentional or intentional errors in implementation of the TOE design may occur, leading to flaws that may be exploited by a malicious user or program.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain access to data or TOE resources.
T.POOR_TEST	Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.
T.REPLAY	A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes.

T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (i.e., system memory, persistent storage, and processing time) via a resource exhaustion denial of service attack.
T.TSF_COMPROMISE	A malicious process or user may cause TSF data or executable code to inappropriately accessed (viewed, modified, or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access (view, modify, delete) to user data.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

3.4 Security Policy

69 Policy statements whose enforcement must be provided by the operating system's security mechanisms:

P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZATION	The TOE must limit the extent of each user's abilities in accordance with the TSP.
P.AUTHORIZED_USERS	Only those users who have been authorized to access the information within the TOE may access the TOE.

P.I_AND_A	All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.
P.INDEPENDENT_TESTING	The TOE must undergo independent testing as part of an independent vulnerability analysis.
P.NEED_TO_KNOW	The TOE must limit the access to the information in protected resources to those authorized users who have a need to know that information.
P.RATINGS_MAINTENANCE	Procedures to maintain the TOE's rating must be in place to maintain the TOE's rating once it is evaluated.
P.REMOTE_ADMIN_ACCESS	Remote administration shall be securely managed by the TOE.
P.ROLES	The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.
P.SYSTEM_INTEGRITY	The TOE shall provide the ability to periodically validate its correct operation and, with the help of administrators, it must be able to recover from any errors that are detected.
P.TRACE	The TOE shall provide the ability to review the actions of individual users.
P.TRUSTED_RECOVERY	Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained
P.VULNERABILITY_ANALYSIS	The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.

3.5 Security Usage Assumptions

70 Assumptions about the use of the IT operating system:

A.PHYSICAL	It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

4. Security Objectives

71 This section defines the security objectives for the TOE and its environment. These objectives are suitable to counter all identified threats and cover all identified organizational security policies and assumptions. The TOE security objectives are identified with "O." appended to the beginning of the name and the environment objectives are identified with "OE." appended to the beginning of the name.

4.1 TOE Security Objectives

O.ACCESS	The TOE will ensure that users gain only authorized access to it and to resources that it controls.
O.ACCESS_HISTORY	The TOE will display information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_ROLE	The TOE will provide an administrator roles to isolate administrative actions.
O.ADMIN_GUIDANCE	The TOE will provide authorized administrators with the necessary information for secure management of the TOE.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O. AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O. AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.
O.CHANGE_MANAGEMENT	The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.

O.DISCRETIONARY_ACCESS	The TOE will control accesses to resources based upon the identity of users and groups of users.
O.DISCRETIONARY_USER_CONTROL	The TOE will allow authorized users to specify which resources may be accessed by which users and groups of users.
O.DISPLAY_BANNER	The system will display an advisory warning regarding use of the TOE.
O.INSTALL_GUIDANCE	The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.PENETRATION_TESTING	The TOE will undergo independent penetration testing to demonstrate that the design and implementation of the TOE prevents users from violating the TOE's security policy.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RATINGS_MAINTENANCE	Procedures to maintain the TOE's rating will be documented.
O.RECOVERY	Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.
O.REPLAY_DETECTION	The TOE will provide a means to detect and reject the replay of authentication data, as well as, TSF data and security attributes.
O.RESIDUAL_INFORMATION	The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.

O.RESOURCE_SHARING	The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (i.e., system memory, persistent storage, and processing time).
O.REFERENCE_MONITOR	The operating system will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
O.SOUND_DESIGN	The TOE will be designed using sound design principles and techniques. The TOE design, design principles and techniques, will be adequately and accurately documented.
O.SOUND_IMPLEMENTATION	The implementation of the TOE will be an accurate instantiation of its design.
O.FUNCTIONAL TESTING	The TOE will undergo independent testing and includes test scenarios and results.
O.TRAINED_USERS	The TOE will provide authorized users with the necessary guidance for secure use of the TOE, to include secure sharing of user data.
O.TRUSTED_SYSTEM_OPERATION	The IT operating system will function in a manner that maintains IT security.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of the user.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.
O.VULNERABILITY_ANALYSIS	The TOE will undergo appropriate vulnerability analysis for vulnerabilities that are obvious.

4.2 Environment Security Objectives

OE.PHYSICAL	Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.

5. Security Functional Requirements

72 This section contains detailed security functional requirements for the operating systems' trusted security functions (TSF) of general-purpose COTS operating systems. These requirements are applied against the operating system in conjunction with the underlying hardware that supports it. The requirements contained in this section are either selected from Part 2 of the CC or have been explicitly stated (with short names ending in "_EXP"). Table 5.1 lists the explicit functional requirements in this section.

Table 5.1 - Explicit Functional Requirements

Explicit Component	Component Behavior Name
FPT_TRC_EXP.1	Internal TSF Data Consistency

5.1 Security Audit (FAU)

5.1.1 Security Audit Data Generation (FAU_GEN)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 **Refinement:** The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

b) All auditable events **listed in Table 5.2**;

c) **All other security relevant auditable events** for the minimal level of audit;

Application Note: For other security relevant functions that are not included in this PP, the ST author defines a basic level of audit.

d) Start-up and shutdown of the operating system; and

e) Uses of special permissions (e.g., those often used by authorized administrators) that circumvent the access control policies.

Table 5.2 - Auditable Events

Requirement	Audit events prompted by requirement
Audit Data Generation (FAU_GEN.1)	(none)
User Identity Association (FAU_GEN.2)	(none)
Audit Review (FAU_SAR.1)	• Opening the audit trail.
Restricted Audit Review (FAU_SAR.2)	• Unsuccessful attempts to read information from the audit records

Selectable Audit Review (FAU_SAR.3)	(none)
Selective Audit (FAU_SEL.1)	• All modifications to the audit configuration that occur while the audit collection functions are operating.
Protected Audit Trail Storage (FAU_STG.1)	(none)
Action in case of possible audit data loss (FAU_STG.3)	• Actions taken due to exceeding of a threshold.
Subset Access Control (FDP_ACC.1)	(none)
Security Attribute Based Access Control (FDP_ACF.1)	• All requests to perform an operation on an object covered by the SFP.
Basic Internal Transfer Protection (FDP_ITT.1)	• All attempts to transfer user data, including identification of the protection method used and any error that occurred.
Subset Residual Information Protection (FDP_RIP.1)	(none)
Authentication Failure Handling (FIA_AFL.1)	• The reaching of the threshold for the unsuccessful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).
User Attribute Definition (FIA_ATD.1)	(none)
Verification of Secrets (FIA_SOS.1)	• Rejection or acceptance by the TSF of any tested secret.
Timing of Authentication (FIA_UAU.1)	• All use of the authentication mechanism
Re-authenticating (FIA_UAU.6)	• All re-authentication attempts.
Protected Authentication Feedback (FIA_UAU.7)	(none)
Timing of Identification (FIA_UID.1)	• All use of the user identification mechanism, including the user identity provided.
User-Subject Binding (FIA_USB.1)	• Success and failure of binding of user security attributes to a subject (e.g. success and failure to create of a subject).
Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1(1))	• All modifications in the behavior of the functions in the TSF.
Management of Security Functions Behavior (for authentication data) (FMT_MOF.1(2))	• All modifications in the behavior of the functions in the TSF.
Management of Security Attributes (FMT_MSA.1)	• All modifications of the values of security attributes.

Secure Security Attributes (FMT_MSA.2)	• All offered and rejected values for a security attribute.
Static Attributes Initialization (FMT_MSA.3)	• Modifications of the default setting of permissive or restrictive rules. • All modifications of the initial values of security attributes.
Management of TSF Data (for general TSF data) (FMT_MTD.1(1))	• All modifications of the values of TSF data.
Management of TSF Data (for audit data) (FMT_MTD.1(2))	• All modifications of the values of audit data.
Management of TSF Data (for previously written audit records) (FMT_MTD.1(3))	(none)
Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1(4))	• All initializations of the values of user security attributes.
Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1(5))	• All modifications of the values of user security attributes.
Management of TSF Data (for modification of authentication data) (FMT_MTD.1(6))	• All actions associated with modifications of the values of authentication data.
Management of TSF Data (for reading of authentication data) (FMT_MTD.1(7))	(none)
Revocation (to authorized administrators) (FMT_REV.1(1))	• All attempts to revoke security attributes.
Revocation (to owners and authorized administrators) (FMT_REV.1(2))	• All attempts to revoke security attributes.
Time-Limited Authorization (FMT_SAE.1)	• Specification of the expiration time for an attribute • Action taken due to attribute expiration.
Security Roles (FMT_SMR.1)	• Modifications to the group of users that are part of a role.
Abstract Machine Testing (FPT_AMT.1)	• Execution of the tests of the underlying machine and the results of the tests.
Basic Internal TSF Data Transfer Protection (FPT_ITT.1)	(none)
TSF Data Integrity Monitoring (FPT_ITT.3)	• Detection of modification of TSF data

Manual Recovery (FPT_RCV.1)	• The fact that a failure or service discontinuity occurred. • Resumption of the regular operation. • Type of failure or service discontinuity
Non-Bypassability of the TSF (FPT_RVM.1)	(none)
TSF Domain Separation (FPT_SEP.1)	(none)
Reliable Time Stamps (FPT_STM.1)	• Changes to the time.
Internal TSF Data Consistency (FPT_TRC_EXP.1)	• Any detection of inconsistency between TSF data.
Maximum Quotas (for persistent storage) (FRU_RSA.1(1))	• Rejection of allocation operation due to persistent storage limits.
Maximum Quotas (for system memory) (FRU_RSA.1(2))	• Rejection of allocation operation due to percentage of system memory limits.
Maximum Quotas (for processing time) (FRU_RSA.1(3))	• Rejection of allocation operation due to processing time limits.
Limitation on scope of selectable attributes (FTA_LSA.1)	• All attempts at selecting a session security attribute.
Basic limitation on multiple concurrent sessions (FTA_MCS.1)	• Rejection of a new session based on the limitation of multiple concurrent sessions.
TSF-Initiated Session Locking (FTA_SSL.1)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.
User-Initiated Locking (FTA_SSL.2)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.
Default TOE Access Banners (FTA_TAB.1)	(none)
TOE Access History (FTA_TAH.1)	(none)
TOE Session Establishment (FTA_TSE.1)	• All attempts at establishment of a user session.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

Application Note: "Subject identity" means user identity associated with the subject.

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST,

• the name of the object; and

• for changes to TSF data, (except for authentication data) the new and old value of the data.

Application Note: TSF data includes access control attributes, user security attributes, definition of roles, and user authorizations.

Application Note: Other audit relevant information associated with security-relevant functions not included in this PP should be included within the audit records.

5.1.1.2 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

Application Note: For failed login attempts no user association is required because the user is not under TSF control until after a successful identification/authentication.

5.1.2 Security Audit Review (FAU_SAR)

5.1.2.1 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide **authorized administrators** with the capability to read **all audit information** from the audit records.

Application Note: For a distributed system, the authorized administrator should be able to read all audit information within the TOE.

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information **using a tool to access the audit trail.1**

Application Note: The tool provides a means to easily and efficiently review the audit data. It is expected (yet not necessary) that the tool satisfying this requirement will also satisfy the FAU_SAR.3 and FAU_SEL.1 requirements.

5.1.2.2 Restricted Audit Review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.1.2.3 Selectable Audit Review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform searches and sorting of audit data based on **the following attributes:**

a) user identity,

b) object identity,

c) date of the event,

d) time of the event,

e) type of event,

f) success of auditable security events, and

g) failure of auditable security events.

5.1.3 Security Audit Event Selection (FAU_SEL)

5.1.3.1 Selective Audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) object identity,

b) user identity,

c) host identity,

d) event type,

e) success of auditable security events, and

f) failure of auditable security events.

5.1.4 Security Audit Event Storage (FAU_STG)

5.1.4.1 Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to prevent modifications to the audit records.

Application Note: In order to reduce the performance impact of audit generation, audit records are often temporarily buffered in memory before being written to the disk. In such implementations, these buffered records will be lost if the operation of the TOE is interrupted by hardware or power failures. The developer should document the expected loss in such circumstances and show that it has been minimized.

5.1.4.2 Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1: **Refinement:** The TSF shall **notify an authorized administrator of the possible audit data loss** if the audit trail exceeds **an authorized administrator selectable, pre-defined limit**. **2**

5.2 User Data Protection (FDP)

5.2.1 Access Control Policy (FDP_ACC)

5.2.1.1 Subset Access Control (FDP_ACC.1)

FDP_ACC.1.1 The TSF shall enforce the **Discretionary Access Control policy** on **all subjects, all operating system controlled files, and all operations among them**.

Application Note: Operating system controlled files include all communications mechanisms – for internal or external communications – that are implemented as objects within the file system.

5.2.2 Access Control Functions (FDP_ACF)

5.2.2.1 Security Attribute Based Access Control (FDP_ACF.1)

FDP_ACF.1.1 **Refinement:** The TSF shall enforce the **Discretionary Access Control policy** to **operating system controlled files** based on:**3**

a) the authorized user identity and group membership(s) associated with a subject and

b) access operations implemented for operating system controlled files.

FDP_ACF.1.2 **Refinement:** The TSF shall enforce the following rules to determine if an operation among subjects and **operating system controlled files** is allowed:**4**

- **The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that operating system controlled files are protected from unauthorized access according to the following ordered rules:**
  1) **If the requested mode of access is denied to that authorized user, deny access.**
  2) **If the requested mode of access is permitted to that authorized user, permit access.**
  3) **If the requested mode of access is denied to every group of which the authorized user is a member, deny access**
  4) **If the requested mode of access is permitted to any group of which the authorized user is a member, grant access**
  5) **Else deny access.**

FDP_ACF.1.3 **Refinement:** The TSF shall explicitly authorize access of subjects to **operating system controlled files** based on the following additional rules:**5**

a) Authorized administrators must follow the above-stated Discretionary Access Control policy, except after taking the following specific actions: *[assignment: list of specific actions]*.

Application Note: This element allows specifications of additional rules for authorized administrators to bypass the Discretionary Access Control policy for system management or maintenance (e.g., system backup).

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following rules: **none**.

5.2.3 Internal TOE Transfer (FDP_ITT)

5.2.3.1 Basic Internal Transfer Protection (FDP_ITT.1)

FDP_ITT.1.1 The TSF shall enforce the **Discretionary Access Control policy** to prevent the disclosure and modification of user data when it is transmitted between physically-separated parts of the TOE.

Application Note: If not physically protected (see A.PHYSICAL), other protection mechanisms that prevent disclosure and modification of user data include link encryption, application-level protection (SHTTP), or some other mechanism described in the ST.

5.2.4 Residual Information Protection (FDP_RIP)

5.2.4.1 Subset Residual Information Protection (FDP_RIP.1)

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon *[selection: allocation of the resource to, deallocation of the resource from]* **shared memory and operating system controlled files**.[6]

5.3 Identification and Authentication (FIA)

5.3.1 Authentication Failures (FIA_AFL)

5.3.1.1 Authentication Failure Handling (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when **an authorized administrator configurable positive integer of consecutive** unsuccessful authentication attempts occur related to **any authorized user authentication process**.

FIA_AFL.1.2 **Refinement:** When the defined number of consecutive unsuccessful authentication attempts has been met or surpassed, the TSF shall:

a) For all administrator accounts, disable the account for an authorized administrator configurable time period;

b) For all other accounts, disable the user logon account until it is re-enabled by the authorized administrator.

c) For all disabled accounts, respond with an "account disabled" message without attempting any type of authentication.

Application Note: "Consecutive unsuccessful authentication attempts" is the total number of unsuccessful attempts that occur, in order, prior to a successful authentication attempt. For distributed systems, the TOE must reconcile unsuccessful attempts across nodes in accordance with FPT_TRC_EXP.1.

5.3.2 User Attribute Definition (FIA_ATD)

5.3.2.1 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) unique identifier;

b) group memberships;

c) authentication data; and

d) security-relevant roles (see FMT_SMR.1)

e) *[Assignment: Any other security-relevant authorizations or attributes (e.g., privilege)].*

Application Note: Group membership may be expressed in a number of ways: a list per user specifying to which groups the user belongs, a list per group which includes which users are members, or implicit association between certain user identities and certain groups.

Application Note: A TOE may have two forms of user and group identities, a text form and a numeric form, which have a unique mapping between the representations.

Application Note: It is possible that the notion of privilege is tied to the security-relevant roles (item d).

5.3.3 Specification of Secrets (FIA_SOS)

5.3.3.1 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following:**

a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 5×10^{15};

Application Note: This can be achieved with a password of eight characters, assuming an alphabet of 92 characters.

b) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

Application Note: The ST specifies the method of authentication. Where authentication is provided by a password mechanism, the ST shows that the restrictions upon passwords (length, alphabet, and other characteristics) result in a password space conforming to item (a) above.

Where authentication is provided by a mechanism other than passwords, the ST shows the authentication method has a low probability that authentication data can be forged or guessed.

5.3.4 User Authentication (FIA_UAU)

5.3.4.1 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1 **Refinement:** The TSF shall allow **read access to *public objects*** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 **Refinement:** The TSF shall require each user to be successfully authenticated **(i.e., an exact match between the user's entered data and the stored TSF authentication data)** before allowing any other TSF-mediated actions on behalf of that user.

Application Note: The entire entered user's authentication data must exactly match the entire stored data. No other parameters such as length of password should be used to short-circuit the authentication verification.

5.3.4.2 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 **Refinement:** The TSF shall re-authenticate the user **when changing authentication data**.***1***

5.3.4.3 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the user while the authentication is in progress.

Application Note: "Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user, which may provide any indication of the authentication data.

5.3.5 User Identification (FIA_UID)

5.3.5.1 Timing of Identification (FIA_UID.1)

FIA_UID.1.1 **Refinement:** The TSF shall allow **read access to *[assignment: list of public objects]*** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.3.6 User-Subject Binding (FIA_USB)

5.3.6.1 User-Subject Binding (FIA_USB.1)

FIA_USB.1.1 **Refinement:** The TSF shall associate the **following** user security attributes with subjects acting on behalf of that user:

a) The unique user identity that is associated with auditable events;

b) The user identity or identities that are used to enforce the Discretionary Access Control Policy;

Application Note: The DAC and audit policies require that each subject acting on behalf of a user has a user identity associated with the subject. While this identity is typically the one used at the time of identification to the system, the DAC policy enforced by the TSF may include provisions for making access decisions based upon a different user identity, such as the "set user ID (su)" command in UNIX.

c) The group identity or identities that are used to enforce the Discretionary Access Control Policy;

d) The user's authorized roles.

Application Note: The attributes listed in FIA_USB.1 should be comparable to those listed in FIA_ATD.1.

5.4 Security Management (FMT)

5.4.1 Management of Functions in TSF (FMT_MOF)

5.4.1.1 Management of Security Functions Behavior (for specification of audited events) (FMT_MOF.1(1))

FMT_MOF.1.1(1) **Refinement:** The TSF shall restrict the ability to disable and enable the **audit** functions **and to specify which events are to be audited (see FAU_SEL.1.1)** to **the authorized administrators.**

Application Note: To "specify" means the ability to select what events will be audited.

5.4.1.2 Management of Security Functions Behavior (for authentication data) (FMT_MOF.1(2))

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to **manage the values of security attributes associated with user authentication data** to **authorized administrators.**

Application Note: The word "manage" includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. Security attributes associated with user authentication data include password length, expiration, history, etc.

5.4.2 Management of Security Attributes (FMT_MSA)

5.4.2.1 Management of Security Attributes (FMT_MSA.1)

FMT_MSA.1.1 **Refinement:** The TSF shall restrict the ability to **change** **the value of the operating system controlled files'** security attributes **to authorized administrators and owners of the controlled files.**[9]

5.4.2.2 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

Application Note: Valid implies that the values fall within an appropriate range for that attribute (e.g., the password length attribute must be a non-negative integer).

FMT_MSA.3.1 The TSF shall enforce the **Discretionary Access Control policy** to provide restrictive default values for security attributes that are used to enforce the SFP.

Application Note: The TOE must provide protection by default for all objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects.

FMT_MSA.3.2 **Refinement:** The TSF shall allow the **authorized administrator** to specify alternative initial values to override the default values when an **operating system controlled file** is created.[10]

5.4.3 Management of TSF Data (FMT_MTD)

5.4.3.1 Management of TSF Data (for general TSF data) (FMT_MTD.1(1))

FMT_MTD.1.1(1) The TSF shall restrict the ability to **manage** the **security-relevant TSF data except for audit records, user security attributes, and authentication data** to **the authorized administrator**.

Application Note: The word "manage" includes but is not limited to create, initialize, change default, modify, delete, clear, append, and query. Security attributes associated with user authentication data include password length, password expiration, password history, etc. The restrictions for audit records, user security attributes, and authentication data are specified below.

5.4.3.2 Management of TSF Data (for audit data) (FMT_MTD.1(2))

FMT_MTD.1.1(2) The TSF shall restrict the ability to query, delete, and clear the **audit records** to **authorized administrators.**

5.4.3.3 Management of TSF Data (for previously written audit records) (FMT_MTD.1(3))

FMT_MTD.1.1(3) **Refinement:** The TSF shall **prevent** modification of **previously written audit records**.[11]

5.4.3.4 Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1(4))

FMT_MTD.1.1(4) The TSF shall restrict the ability to **initialize user security attributes** to **authorized administrators**.

5.4.3.5 Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1(5))

FMT_MTD.1.1(5) The TSF shall restrict the ability to modify **user security attributes, other than authentication data,** to **authorized administrators.**

5.4.3.6 Management of TSF Data (for modification of authentication data) (FMT_MTD.1(6))

FMT_MTD.1.1(6) The TSF shall restrict the ability to modify **authentication data** to **authorized administrators and users authorized to modify their own authentication data.**

5.4.3.7 Management of TSF Data (for reading of authentication data) (FMT_MTD.1(7))

FMT_MTD.1.1(7) **Refinement:** The TSF shall **prevent reading of authentication data**.[12]

5.4.4 Revocation (FMT_REV)

5.4.4.1 Revocation (to authorized administrators) (FMT_REV.1(1))

FMT_REV.1.1(1) The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to **authorized administrators**.

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2(1) **Refinement:** The TSF shall enforce the **immediate revocation of security-relevant authorizations.**[13]

Application Note: Security-relevant authorizations include the ability of authorized users to log in or perform privileged operations. An example of revoking a security-relevant authorization is the deletion of a user account upon which system access is immediately terminated).

5.4.4.2 Revocation (to owners and authorized administrators) (FMT_REV.1(2))

FMT_REV.1.1 (2) **Refinement:** The TSF shall restrict the ability to revoke security attributes **of operating system controlled files** within the TSC to **owners and authorized administrators**.[14]

Application Note: The term "revoke security attributes" means "change attributes so that access is revoked".

FMT_REV.1.2 (2) **Refinement:** The TSF shall enforce the **revocation of access rights associated with operating system controlled files when an access check is made.**[15]

Application Note: The state where access checks are made determines when the access control policy enforces revocation. The access control policy may include immediate or delayed revocation. The access rights are considered to have been revoked when all subsequent access control decisions made by the TSF use the new access control information. In cases where a previous access control decision was made to permit an operation, it is not required that every subsequent operation make an explicit access control decision.

5.4.5 Security Attribute Expiration (FMT_SAE)

5.4.5.1 Time-Limited Authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for **authorized user authentication data** to **the authorized administrator**.

FMT_SAE.1.2 **Refinement:** The TSF shall be able to **lock out the associated authorized user account** after the expiration time has passed.[16]

5.4.6 Security Management Roles (FMT_SMR)

5.4.6.1 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the role of **authorized administrator**.

Application Note: Any user that is authorized to modify the TOE such that the DAC policy is bypassed is by definition, an authorized administrator. The TOE may provide multiple administrator roles (audit administrator, security administrator, etc).

FMT_SMR.1.2 **Refinement:** The TSF shall be able to associate **authorized** users with roles.

5.5 Protection of the TOE Security Functions (FPT)

5.5.1 Underlying Abstract Machine Test (FPT_AMT)

5.5.1.1 Abstract Machine Testing (FPT_AMT.1)

FPT_AMT.1.1 **Refinement:** The TSF shall run a suite of tests during the initial start-up or **at the request of an authorized administrator** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the **software portions of the** TSF.

Application Note: The test suite need only cover aspects of the underlying abstract machine on which the TSF relies to implement required functions, including domain separation.

5.5.2 Internal TOE TSF Data Transfer (FPT_ITT)

5.5.2.1 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1 **Refinement:** The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE.

5.5.2.2 TSF Data Integrity Monitoring (FPT_ITT.3)

FPT_ITT.3.1 **Refinement:** The TSF shall be able to detect modification, **insertion** and **replay** of TSF data transmitted between separate parts of the TOE.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions:

a) reject data

b) audit event

c) [assignment: specify the action to be taken].

Application Note: Additional actions ST author might consider are: retransmission of data and, an alarm after reaching a retransmission threshold.

5.5.3 Trusted Recovery (FPT_RCV)

5.5.3.1 Manual Recovery (FPT_RCV.1)

FPT_RCV.1.1 **Refinement:** After a failure or service discontinuity, **that may lead to a violation of the TSP,** the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

5.5.4 Reference Mediation (FPT_RVM)

5.5.4.1 Non-Bypassability of the TSF (FPT_RVM.1)

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.5.5 Domain Separation (FPT_SEP)

5.5.5.1 TSF domain separation (FPT_SEP.1)

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

5.5.6 Time Stamps (FPT_STM)

5.5.6.1 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Application Note: A time stamp includes the correct date and time.

5.5.7 Internal TOE TSF Data Replication Consistency (FPT_TRC)

5.5.7.1 **Explicit: Internal TSF Data Consistency (FPT_TRC_EXP.1)**

FPT_TRC_EXP.1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state in a timely manner.

Application Note: In general, it is impossible to achieve complete, constant consistency of TSF data that is distributed to remote portions of a TOE because distributed portions of the TSF may be active at different times or disconnected from one another. This requirement attempts to address this situation in a practical manner by acknowledging that there will be TSF data inconsistencies but that they will be corrected without undue delay. For example, a TSF could provide timely consistency through periodic broadcast of TSF data to all TSF nodes maintaining replicated TSF data. Another example approach is for the TSF to provide a mechanism to explicitly probe remote TSF nodes for inconsistencies and respond with action to correct the identified inconsistencies.

5.6 Resource Utilization (FRU)

5.6.1 Resource Allocation (FRU_RSA)

5.6.1.1 Maximum Quotas (for disk space) (FRU_RSA.1(1))

FRU_RSA.1.1(1) The TSF shall enforce maximum quotas of the following resources: **portion of disk space** that individual **authorized** users can use simultaneously.

5.6.1.2 Maximum Quotas (for system memory) (FRU_RSA.1(2))

FRU_RSA.1.1(2) The TSF shall enforce maximum quotas of the following resources: **portion of system memory** that individual **authorized** users can use simultaneously.

5.6.1.3 Maximum Quotas (for processing time) (FRU_RSA.1(3))

FRU_RSA.1.1(3) The TSF shall enforce maximum quotas of the following resources: **portion of processing time** that subjects can use over a specified period of time.

Application Note: The algorithm to determine portion of time can be based on many factors (e.g., number of users, relative priority of users, availability of resources to users).

5.7 TOE Access (FTA)

5.7.1 Limitation on scope of selectable attributes (FTA_LSA)

5.7.1.1 Limitation on scope of selectable attributes (FTA_LSA.1)

FTA_LSA.1.1 **Refinement**: The TSF shall restrict the scope of **roles and user privileges** based on **location, time, and day**.[17]

Application Note: The intent of this requirement is to allow or disallow the assumption of roles or the effectiveness of user privileges based on the location where the session was established or the date/time of session establishment.

Application Note: "Location" refers to what ever means the TOE uses to identify a point of entry for interactive user session establishment. The adequacy of this means is determined by other requirements (e.g., FPT_SEP, AVA_VLA).

5.7.2 Limitation on multiple concurrent sessions (FTA_MCS)

5.7.2.1 Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1 **Refinement:** The TSF shall **enforce a** maximum number of concurrent **interactive** sessions per user.[18]

FTA_MCS.1.2 **Refinement:** The TSF shall allow **an authorized administrator to set the maximum number of concurrent interactive** sessions per user.[19]

Application Note: "Concurrent" refers to any specific synchronization as defined in the internal TSF data consistency requirement FPT_TRC_EXP.1.1. Enforcement of the requirement is at every synchronization.

5.7.3 Session Locking (FTA_SSL)

5.7.3.1 TSF-Initiated Session Locking (FTA_SSL.1)

FTA_SSL.1.1The TSF shall lock an interactive session after *[assignment: a time interval of user inactivity]* by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 **Refinement:** The TSF shall require the **user to re-authenticate** prior to unlocking the session.[20]

5.7.3.2 User-Initiated Locking (FTA_SSL.2)

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 **Refinement:** The TSF shall require the **user to re-authenticate** prior to unlocking the session.[21]

5.7.4 TOE Access Banners (FTA_TAB)

5.7.4.1 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 **Refinement:** Before establishing a user session, the TSF shall display an **authorized-administrator specified** advisory **notice and consent** warning message regarding unauthorized use of the TOE.

5.7.5 TOE Access History (FTA_TAH)

5.7.5.1 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1 **Refinement:** Upon successful **interactive** session establishment, the TSF shall display **to the authorized user** the date and time of **that authorized user's** last successful **interactive** session establishment.

Upon successful **interactive** session establishment, the TSF shall display **to the authorized user** the date and time of the last unsuccessful attempt and the number of unsuccessful attempts at **interactive** session establishment **for that user identifier** since the last successful **interactive** session establishment.

Application Note: In both of the above elements, for distributed systems, date and time needs to be accurate to the degree required by FPT_TRC_EXP.1.

FTA_TAH.1.3 **Refinement:** The TSF shall not erase the access history information from the **authorized** user interface without giving the **authorized** user the opportunity to review the information.

5.7.6 TOE Session Establishment (FTA_TSE)

5.7.6.1 TOE Session Establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **location, time, and day**.

End Notes

This section records the functional requirements where deletions of Common Criteria text were performed.

---

1 A deletion of CC text was performed in FAU_SAR.1.2. Rationale: The word "user" was deleted to replace it with "authorized administrator". By default, authorized administrators are the only users with read access to audit records unless granted explicit read-access (FAU_SAR.2).

FAU_SAR.1.2 **Refinement:** The TSF shall provide the audit records in a manner suitable for the ~~user~~ **authorized administrator** to interpret the information **using a tool to access the audit trail**.

2 A deletion of CC text was performed in FAU_STG.3.1. Rationale: The word "take" was deleted for clarity and better flow of the requirement.

FAU_STG.3.1 - **Refinement:** The TSF ~~take~~ shall **notify an authorized administrator of the possible audit data loss** if the audit trail exceeds **an authorized administrator selectable, pre-defined limit**.

3 A deletion of CC text was performed in FDP_ACF.1.1. Rationale: The word "objects" was deleted and replaced with "operating system controlled files" to refine the scope of SFP controlled objects. In this PP, the DAC policy will only be enforced on the OS controlled files as identified in FDP_ACC.1.

FDP_ACF.1.1 **Refinement**: The TSF shall enforce the **Discretionary Access Control policy** to ~~objects~~ **operating system controlled files** based on:

a) the authorized user identity and group membership(s) associated with a subject; and

b) access operations implemented for operating system controlled files.

4 A deletion of CC text was performed in FDP_ACF.1.2. Rationale: The words "controlled" and "controlled objects" were deleted. There is no need to specify "controlled" subjects since it has not been defined that way. "Controlled objects" was replaced with "operating system controlled files" to refine the scope of SFP controlled objects. In this PP, the DAC policy will only be enforced on the OS controlled files as identified in FDP_ACC.1.

FDP_ACF.1.2 **Refinement:** The TSF shall enforce the following rules to determine if an operation among controlled subjects and ~~controlled objects~~ **operating system controlled files** is allowed…

5 A deletion of CC text was performed in FDP_ACF.1.3. Rationale: The word "objects" was deleted and replaced with "operating system controlled files" to refine the scope of SFP controlled objects. In this PP, the DAC policy will only be enforced on the OS controlled files as identified in FDP_ACC.1.

FDP_ACF.1.3 **Refinement:** The TSF shall explicitly authorize access of subjects to ~~objects~~ **operating system controlled files** based on the following additional rules…

6 A deletion of CC text was performed in FDP_RIP.1.1. Rationale: The words "the following objects" were deleted for better clarity and flow on the element.

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon [**selection:** allocation of the resource to, deallocation of the resource from] ~~the following objects~~ **shared memory and operating system controlled files**.

7 A deletion of CC text was performed in FIA_UAU.6.1. Rationale: The words "under the conditions" were deleted for better clarity and flow on the element.

FIA_UAU.6.1 **Refinement:** The TSF shall re-authenticate the user ~~under the conditions~~ **when changing authentication data**.

8 A deletion of CC text was performed in FMT_MOF.1.1(2). Rationale: The words "the functions" were deleted for clarity and better flow of the requirement.

FMT_MOF.1.1(2) **Refinement:** The TSF shall restrict the ability to **manage** ~~the functions~~ **the values of security attributes associated with user authentication data** to **authorized administrators.**

9 A deletion of CC text was performed in FMT_MSA.1.1. Rationale: The words " enforce the [assignment: access control SFP, information flow control SFP]]" was deleted for clarity and better flow of the requirement.

FMT_MSA.1.1 **Refinement:** The TSF shall ~~enforce the [assignment: access control SFP, information flow control SFP] to~~ restrict the ability to change the value of the **operating system controlled files'** security attributes to authorized administrators and owners of the object.

10 A deletion of CC text was performed in FMT_MSA.3.2. Rationale: The words "object or information" were deleted and replaced with "operating system controlled files" to refine the scope of SFP controlled objects. In this PP, the DAC policy will only be enforced on the OS controlled files as identified in FDP_ACC.1.

FMT_MSA.3.2 **Refinement:** The TSF shall allow the authorized administrator to specify alternative initial values to override the default values when an ~~object or information~~ **operating system controlled file** is created.

11 A deletion of CC text was performed in FMT_MTD.1.1(3). Rationale: The words "restrict" and the assignment "to [assignment: the authorized identified roles]." were deleted for clarity and better flow of the requirement.

FMT_MTD.1.1(3) **Refinement:** The TSF shall **prevent** ~~restrict~~ the ability to modify **previously written audit records** to ~~[assignment: the authorized identified roles].~~

12 A deletion of CC text was performed in FMT_MTD.1.1(7). Rationale: The words "restrict" and the assignment "to [assignment: the authorized identified roles]." were deleted for clarity and better flow of the requirement.

FMT_MTD.1.1(7) **Refinement:** The TSF shall **prevent** ~~restrict~~ the ability to **reading of authentication data** ~~to [assignment: the authorized identified roles].~~

13 A deletion of CC text was performed in FMT_REV.1.2 (1). Rationale: The word "rules" was deleted for clarity and better flow of the requirement.

FMT_REV.1.2(1) **Refinement:** The TSF shall enforce the ~~rules~~ **immediate revocation of security-relevant authorizations**.

14 A deletion of CC text was performed in FMT_REV.1.1 (2). Rationale: The words "associated with" were deleted for clarity and better flow of the requirement.

FMT_REV.1.1 (2) **Refinement:** The TSF shall restrict the ability to revoke security attributes ~~associated with~~ **of operating system controlled files** within the TSC to **owners and authorized administrators.**

15 A deletion of CC text was performed in FMT_REV.1.2 (2). Rationale: The word "rules" was deleted for clarity and better flow of the requirement.

FMT_REV.1.2 (2) **Refinement:** The TSF shall enforce the ~~rules~~ **revocation of access rights associated with operating system controlled files when an access check is made**.

16 A deletion of CC text was performed in FMT_SAE.1.2. Rationale: The words " For each of these security attributes," and "for the indicated security attribute**"** were deleted for clarity and better flow of the requirement.

FMT_SAE.1.2 **Refinement**, ~~For each of these security attributes~~, The TSF shall be able to **lock out the associated authorized user account** after the expiration time ~~for the indicated security attribute~~ has passed.

17 A deletion of CC text was performed in FTA_LSA.1.1. Rationale: The words "the session security attributes" were deleted for clarity and better flow of the requirement.

FTA_LSA.1.1 **Refinement**: The TSF shall restrict the scope of ~~the session security attributes~~ **roles and user privileges** based on **location, time, and day**.

18 A deletion of CC text was performed in FTA_MCS.1.1. Rationale: The words "restrict the" and "that belong to the same" were deleted for clarity and better flow of the requirement.

FTA_MCS.1.1 **Refinement:** The TSF shall ~~restrict the~~ enforce a maximum number of concurrent **interactive** sessions ~~that belong to the same~~ per user.

19 A deletion of CC text was performed in FTA_MCS.1.2. Rationale: The words "enforce, by default, a limit of" were deleted to refine the requirement to allow for a settable limit of sessions per user.

FTA_MCS.1.2 **Refinement:** The TSF shall ~~enforce, by default, a limit of~~ allow **an administrator to set the maximum number of concurrent interactive** sessions per user.

20 A deletion of CC text was performed in FTA_SSL.1.2. Rationale: The words "following events to occur" were deleted for clarity and better flow of the requirement.

FTA_SSL.1.2 **Refinement:** The TSF shall require the ~~following events to occur~~ **user to re-authenticate** prior to unlocking the session.

21 A deletion of CC text was performed in FTA_SSL.2.2. Rationale: The words "following events to occur" were deleted for clarity and better flow of the requirement.

FTA_SSL.2.2 **Refinement:** The TSF shall require the ~~following events to occur~~ **user to re-authenticate** prior to unlocking the session.

6. Security Assurance Requirements

73 This section contains detailed security assurance requirements for general-purpose COTS operating systems. The requirements contained in this section have been selected from Part 3 of the CC.

74 The combination of assurance components chosen for the intended environment results in an Evaluated Assurance Level 2 Augmented (EAL2+). The chosen augmented assurances are in the areas of configuration management capabilities and scope, implementation representation, security policy modeling, coverage, functional tests, and misuse. These security assurance requirements are summarized in Table 6.1. Note that flaw remediation (ALC_FLR.2) has also been chosen even though the CC those not assign this component to a specific EAL level.

Table 6.1 - Summary of Assurance Components by Evaluation Assurance Level

Assurance Class	Assurance Family	Assurance Components by Evaluation Assurance Level						
		EAL1	**EAL2**	EAL3	EAL4	EAL5	EAL6	EAL7
Configuration Management	ACM_AUT				1	1	2	2
	ACM_CAP	1	2	**3**	4	4	5	5
	ACM_SCP			1	**2**	3	3	3
Delivery and Operation	ADO_DEL		**1**	1	2	2	2	3
	ADO_IGS	1	**1**	1	1	1	1	1
Development	ADV_FSP	1	**1**	1	2	3	3	4
	ADV_HLD		**1**	2	2	3	4	5
	ADV_IMP				**1**	2	3	3
	ADV_INT					1	2	3
	ADV_LLD				1	1	2	2
	ADV_RCR	1	**1**	1	1	2	2	3
	ADV_SPM				**1**	3	3	3
Guidance Documents	AGD_ADM	1	**1**	1	1	1	1	1
	AGD_USR	1	**1**	1	1	1	1	1
Life cycle Support	ALC_DVS			1	1	1	2	2
	ALC_FLR		**(2)**					
	ALC_LCD				1	2	2	3
	ALC_TAT				1	2	3	3
Tests	ATE_COV		1	**2**	2	2	3	3
	ATE_DPT			**1**	1	2	2	3
	ATE_FUN		**1**	1	1	1	2	2
	ATE_IND	1	**2**	2	2	2	2	3
Vulnerability Assessment	AVA_CCA					**1**	2	2
	AVA_MSU			**1**	2	2	3	3
	AVA_SOF		**1**	1	1	1	1	1
	AVA_VLA		**1**	1	2	3	4	4

6.1 Configuration Management (ACM)

6.1.1 CM Capabilities (ACM_CAP)

6.1.1.1 Authorization Controls (ACM_CAP.3)

ACM_CAP.3.1D The developer shall provide a reference for the TOE.

ACM_CAP.3.2D The developer shall use a CM system.

ACM_CAP.3.3D The developer shall provide CM documentation.

ACM_CAP.3.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.3.2C The TOE shall be labeled with its reference.

ACM_CAP.3.3C The CM documentation shall include a configuration list and a CM plan.

ACM_CAP.3.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.3.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.3.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.3.7C The CM plan shall describe how the CM system is used.

ACM_CAP.3.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.3.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.3.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.1.2 CM Scope (ACM_SCP)

6.1.2.1 Problem Tracking CM Coverage (ACM_SCP.2)

ACM_SCP.2.1D The developer shall provide CM documentation.

ACM_SCP.2.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

ACM_SCP.2.2C The CM documentation shall describe how configuration items are tracked by the CM system.

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2 Delivery and Operation (ADO)

6.2.1 Delivery (ADO_DEL)

6.2.1.1 Delivery Procedures (ADO_DEL.1)

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.2 Installation, Generation and Start-up (ADO_IGS)

6.2.2.1 Installation, Generation, and Start-Up Procedures (ADO_IGS.1)

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

6.3 Development Documentation (ADV)

6.3.1 Functional Specification (ADV_FSP)

6.3.1.1 Informal Functional Specification (ADV_FSP.1)

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

ADV_FSP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

6.3.2 High-Level Design (ADV_HLD)

6.3.2.1 Descriptive High-Level Design (ADV_HLD.1)

ADV_HLD.1.1D The developer shall provide the high-level design of the TSF.

ADV_HLD.1.1C The presentation of the high-level design shall be informal.

ADV_HLD.1.2C The high-level design shall be internally consistent.

ADV_HLD.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.1.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

6.3.3 Implementation Representation (ADV_IMP)

6.3.3.1 Subset of the Implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

ADV_IMP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

6.3.4 Representation Correspondence (ADV_RCR)

6.3.4.1 Informal Correspondence Demonstration (ADV_RCR.1)

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.3.5 Security Policy Modeling (ADV_SPM)

6.3.5.1 Informal TOE Security Policy Model (ADV_SPM.1)

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

Application Note: Security policies that can be modeled include descriptions of at least the following security policies: Identification and Authentication, Discretionary Access Control, and Audit.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.4 Guidance Documents (AGD)

6.4.1 Administrator Guidance (AGD_ADM)

6.4.1.1 Administrator Guidance (AGD_ADM.1)

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

Application Note: Administrators of the TOE include the "authorized administrator" role (see FMT_SMR.1).

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.4.2 User Guidance (AGD_USR)

6.4.2.1 User Guidance (AGD_USR.1)

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.5 Life Cycle Support (ALC)

6.5.1 Flaw Remediation (ALC_FLR)

6.5.1.1 Flaw Reporting Procedures (ALC_FLR.2)

ALC_FLR.2.1D The developer shall document the flaw remediation procedures.

ALC_FLR.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

ALC_FLR.2.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.2.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.2.5C The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.2.6C The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6 Testing (ATE)

6.6.1 Coverage (ATE_COV)

6.6.1.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.2 Depth (ATE_DPT)

6.6.2.1 Testing: High-Level Design (ATE_DPT.1)

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

ATE_DPT.1.2E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.3 Functional Tests (ATE_FUN)

6.6.3.1 Functional Testing (ATE_FUN.1)

ATE_FUN.1.1D **Refinement:** The developer shall test the TSF **including stress testing the boundary conditions of all external interfaces** and document the results.

Application Note: Stress testing of boundary conditions must be provided for all external TSF interfaces. However, the testing is not expected to be, nor would it be feasible to be, exhaustive. The test documentation should describe the philosophy of the approach to test the interface boundary conditions and should present evidence that the approach is sufficient.

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.6.4 Independent Testing (ATE_IND)

6.6.4.1 Independent Testing - Sample (ATE_IND.2)

ATE_IND.2.1D The developer shall provide the TOE for testing.

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

6.7 Vulnerability Assessment (AVA)

6.7.1 Misuse (AVA_MSU)

6.7.1.1 Examination of Guidance (AVA_MSU.1)

AVA_MSU.1.1D The developer shall provide guidance documentation.

AVA_MSU.1.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.1.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.1.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.1.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.1.2E The evaluator shall repeat all configuration and installation procedures to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.1.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

6.7.2 Strength of TOE security functions (AVA_SOF)

6.7.2.1 Strength of TOE Security Function Evaluation (AVA_SOF.1)

Application Note: The security functions, for which strength of function claims are made, are identified in section 5.3.3.

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

6.7.3 Vulnerability Analysis (AVA_VLA)

6.7.3.1 Developer Vulnerability Analysis (AVA_VLA.1)

AVA_VLA.1.1D The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.

AVA_VLA.1.2D The developer shall document the disposition of obvious vulnerabilities.

AVA_VLA.1.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.1.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

7. Rationale

75 This section provides the rationale for the selection, creation, and use of security objectives and requirements as defined in sections 4 and 5, respectively.

7.1 Security Objectives derived from Threats

76 Each of the identified threats to security is addressed by one or more security objectives. Table 7.1 below provides the mapping from security objectives to threats, as well as a rationale that discusses how the threat is addressed. Definitions are provided (*in italics*) below each thereat and security objectives so the PP reader can reference these without having to go back to sections 3 and 4.

Table 7.1 – Mapping of Security Objectives to Threats

Threat	Objectives Addressing Threat	Rationale
T.ADMIN_ERROR *An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.*	O.ADMIN_GUIDANCE *The TOE will provide administrators with the necessary information for secure management of the TOE.* O.INSTALL_GUIDANCE *The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.* O.MANAGE *The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.*	Improper or insufficient security policies and mechanisms might be implemented if the administrator is not properly trained. However, if the administrator is provided sufficient guidance for the installation [O.INSTALL_GUIDANCE], configuration and management of the TOE [O.ADMIN_GUIDANCE], the threat that the administrator may incorrectly install, configure, or manage the TOE, in a way that undermines security, is reduced. O.MANAGE also contributes to mitigating this threat by providing the security mechanisms (e.g., tools for reviewing audit data) for administrators to perform TOE administration effectively, and to quickly alert the administrator of ineffective security policies on the TOE.
T.ADMIN_ROGUE *An authorized administrator's intentions may become malicious resulting in user or TSF data being compromised.*	O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	It is important to limit the functionality of administrative roles. If the intentions of an individual in an administrative role become malicious, O.ADMIN_ROLE mitigates this threat by isolating the administrative actions within the role and limiting the functions available to that individual. This objective presumes that separate individuals will be assigned separate distinct roles with no overlap of allowed operations between the roles. Separate roles include an authorized administrator and a cryptographic administrator.

T.AUDIT_COMPROMISE *A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future records from being recorded, thus masking a user's actions.*	OE.PHYSICAL *Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.* O.AUDIT_GENERATION *The TOE will provide administrators with the necessary information for secure management of the TOE.* O.AUDIT_PROTECTION *The TOE will provide the capability to protect audit information.* O.REFERENCE_MONITOR *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.*	O.AUDIT_GENERATION provides the capability to detect and create records of security relevant events. Audit records identify the user responsible for the event and are an important form of evidence that can be used to track an attacker's actions. Tampering with or destruction of audit data by physical means is addressed by OE.PHYSICAL, which provides physical security controls to the TOE environment O.AUDIT_PROTECTION provides the capability to specifically protect audit information from external interference, tampering, or unauthorized disclosure. O.REFERENCE_MONITOR protects the TOE and its resources (including audit data) by ensuring that the security policies implemented by the TOE to protect the audit information are always invoked.
T.POOR_DESIGN *Unintentional or intentional errors in requirements specifications or design of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.*	O.CHANGE_MANAGEMENT *The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.* O.SOUND_DESIGN *The TOE will be designed using sound design principles and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.* O.VULNERABILITY_ANALYSIS *The Toe will undergo appropriate vulnerability analysis and penetration testing by NSA to demonstrate the design and implementation of the TOE does not allow attackers with moderate attack potential to violate the TOE's security policies.*	Intentional or unintentional errors may occur in the requirement specification, design or development of the TOE. To address this threat, O.SOUND_DESIGN requires sound design principles and techniques that help prevent faults in the TOE's design by eliminating errors in the logic. In addition, O.CHANGE_MANAGEMENT addressed this threat by requiring all changes to the TOE and its development evidence be analyzed, tracked and controlled throughout the development cycle. To verify that there are no intentional or unintentional errors introduced in the design. O.VULNERABILITY_ANALYSIS demonstrates that the design of the TOE is resistant to attacks that exercise these designs and development errors.

T.POOR_IMPLEMENTATION *Unintentional or intentional errors in implementation of the TOE may occur, leading to flaws that may be exploited by a malicious user or program.*	O.CHANGE_MANAGEMENT *The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development.* O.PENETRATION_TESTING *The TOE will undergo independent penetration testing to demonstrate that the design and implementation of the TOE prevents users from violating the TOE's security policies.* O.SOUND_IMPLEMENTATION *The implementation of the TOE will be an accurate instantiation of its design.* O.FUNCTIONAL_TESTING *The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.* O.VULNERABILITY_ANALYSIS *The TOE will undergo appropriate vulnerability analysis and penetration testing by NSA to demonstrate the design and implementation of the TOE does not allow attackers with moderate attack potential to violate the TOE's security policies.*	Intentional or unintentional errors may occur when implementing the design of the TOE. To address this threat, O.SOUND_IMPLEMENTATION ensures that the implementation is an accurate representation of the design. To ensure that an accurate representation of the design is maintained, O.CHANGE_MANAGEMENT ensures that all changes to the TOE and its development evidence are analyzed, tracked and controlled throughout the development cycle. To ensure that errors have not been introduced, O.FUNCTIONAL_TESTING validates that the TSF satisfies the security functional requirements. To further demonstrate that vulnerabilities are not present, both O.PENETRATION_TESING and O.VULNERABILITY_ANALYSIS ensure correct implementation of the TOE.
T.MASQUERADE *A malicious user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.*	O.USER_AUTHENTICATION *The TOE will verify the claimed identity of users.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	To address this threat, O.USER_IDENTIFICATION identifies the user as a legitimate user and O.USER_AUTHENTICATION authenticates this user preventing unauthorized users, processes, or external IT entities from masquerading as an authorized entity.

T.POOR_TEST *Lack of or insufficient tests to demonstrate that all TOE security functions operate correctly (including in a fielded TOE) may result in incorrect TOE behavior being undiscovered thereby causing potential security vulnerabilities.*	O.PENETRATION_TEST *The TOE will undergo independent penetration testing to demonstrate that the design and implementation of the TOE prevents users from violating the TOE's security policy.* O.CORRECT_TSF_OPERATION *The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.* O.FUNCTIONAL_TESTING *The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.*	Design analysis determines that a TOE's documented design satisfies its security functional requirements. In order to ensure the TOE's design is correctly realized in its implementation, the appropriate level of functional testing of the TOE's security mechanisms must be performed during the evaluation of the TOE. O.FUNCTIONAL_TESTING ensures that adequate functional testing is performed to demonstrate the TSF satisfies the security functional requirements and the TOE's security mechanisms operate as documented. While functional testing serves an important purpose, it does not ensure the TSFI cannot be used in unintended ways to circumvent the TOE's security policies. O.PENETRATION_TEST addresses this concern by requiring a vulnerability analysis be performed in conjunction with testing that goes beyond functional testing. This objective provides a measure of confidence that the TOE does not contain security flaws that may not be identified through functional testing. While these testing activities are a necessary activity for successful completion of an evaluation, this testing activity does not address the concern that the TOE continues to operate correctly and enforce its security policies once it has been fielded. Some level of testing must be available to authorized users to ensure the TOE's security mechanisms continue to operate correctly once the TOE is fielded. O.CORRECT_TSF_OPERATION ensures that once the TOE is installed at a customer's location, the capability exists that the integrity of the TSF (hardware and software) can be demonstrated, and thus provides end users the confidence that the TOE's security policies continue to be enforced.

T.REPLAY *A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes.*	O.REPLAY_DETECTION *The TOE will provide a means to detect and reject the replay of authentication data, as well as, TSF data and security attributes.*	A common security threat is the interception and replay of security relevant information causing undesirable results. To prevent the negative effects of this threat, the TOE must provide mechanisms to ensure appropriate protection of security relevant data while it is in transit. Specifically, the TOE must detect and prevent the replay of an intercepted copy of protected authentication data as well as protected TSF data, such as security-relevant configuration parameters, that could cause the TOE to enter a state not intended by the TOE security administrator. The TOE objective O.REPLAY_DETECTION addresses this threat by ensuring that transmitted TSF data cannot be captured by a malicious user and resubmitted.
T.RESIDUAL_DATA *A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.*	O.RESIDUAL_INFORMATION *The TOE will ensure that any information contained in a protected resource is not released when the resource is reallocated.*	The sharing of hardware resources such as primary and secondary storage components between users introduces the potential for information flow in violation of the TOE security policy when hardware resources are deallocated from one user and allocated to another. In order to prevent such unintended consequences, the TOE prevents the compromise of the TOE security policy through mechanisms that ensure residual information cannot be accessed after the resource has been reallocated (O.RESIDUAL_INFORMATION). The intent here is to prevent the unauthorized flow of information that would violate the TOE security policy. The intent is not to require explicit scrubbing or overwriting of data prior to reuse of the storage resource. Therefore, the presence of "residual" data in a storage resource is acceptable as long as subsequent users cannot access it such that a violation of the TOE security policy results. Note, however, that the requirements for storage resources which contain critical cryptographic security parameters differ from the requirements for other types of data. Refer to the appropriate threat, objectives, and requirements rationale for a discussion of the requirements for residual data protection involving critical cryptographic security parameters.

T.RESOURCE _EXHAUSTION *A malicious process or user may block others from system resources (i.e., system memory, persistent storage, and processing time) via a resource exhaustion denial of service attack.*	O.RESOURCE_SHARING *The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (e.g., system memory, persistent storage, and processing time).*	The sharing of resources (i.e., system memory, persistent storage, and processing time) between users introduces the potential for a malicious process or user to obstruct users from access to resources via a resource exhaustion denial-of-service attack. O.RESOURCE_SHARING mitigates this threat by requiring the TOE to provide controls to enforce maximum quotas for system memory, persistent storage, and processing time.
T.TSF_COMPROMISE *A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).*	OE.PHYSICAL *Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.* O.REFERENCE_MONITOR *The TOE will maintain a domain for its own executions that protects itself and its resources from external interference, tampering, or unauthorized disclosure.*	The tampering with or destruction of TSF hardware, software, or configuration data via physical means is addressed by the physical security controls present in the TOE environment [OE.PHYSICAL]. O.REFERENCE_MONITOR addresses the threat of tampering with or destruction of TSF hardware, software, or configuration data by other (non-physical) means. It ensures that the TSF maintains a security domain for its own execution that protects it from interference and tampering by untrusted subjects and enforces the separation between the security domains of subjects within the TSC.
T.UNATTENDED_SESSION *A user may gain unauthorized access to an unattended session.*	O.PROTECT *The TOE will provide mechanisms to protect user data and resources.* O.TRAINED_USERS *The TOE will provide authorized users with the necessary guidance for secure use of the TOE, to include secure sharing of user data.*	When an authorized user leaves an active session unattended, an unauthorized user may gain access to the unattended session. O.PROTECT mitigates this threat by providing mechanisms to protect user data and resources from unauthorized access by ensuring that the TSF will lock an interactive session and make the visible contents unreadable after a specified time interval of session inactivity. In addition, the TSF also allows authorized users to lock their interactive session before leaving the session unattended [O.TRAINED_USERS].

T.UNAUTHORIZED _ACCESS *A user may gain unauthorized access (view, modify, delete) to user data.*	OE.PHYSICAL *Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.* O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.ACCESS_HISTORY *The TOE will display information (to authorized users) related to previous attempts to establish a session.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	Unauthorized users may physically access TOE resources. To mitigate this threat, OE.PHYSICAL restricts the physical access only to authorized personnel. Within the computing environment, O.ACCESS restricts all access controls to authorized users based on their user identity. At the same time, O.PROTECT enforces access rules by providing mechanisms to prevent the user data from unauthorized disclosure and modification. O.ACCESS_HISTORY helps users confirm their previously established session or may help detected possible unsuccessful attempts to their account by an unauthorized user.
T.UNIDENTIFIED_ACTIONS *The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.*	O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.* O.ADMIN_GUIDANCE *The TOE will provide administrators with the necessary information for secure management of the TOE.*	The threat of an administrator failing to know about audit events may occur. To mitigate this threat, O.AUDIT_REVIEW provides the capability to selectively view audit information, and alert the administrator of identified potential security violations. If alerted, the administrator needs to acknowledge the message and act according to the guidance [O.ADMIN_GUIDANCE].
T.UNKNOWN_STATE When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.	O.RECOVERY Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.	After a failure, the security condition of the TOE may be unknown. To mitigate this threat O.RECOVERY provides procedures and/or mechanisms to ensure that recovery without a protection compromise is obtained. O.SECURE_STATE provides the mechanisms to verify the correctness of the TSF code and data thus ensuring a secure state after a failure or upon startup.

7.2 Objectives derived from Security Policies

77 Each of the identified security policies implies a set of security objectives to be met. The table below summarizes this mapping; this is then followed by explanatory text of how this mapping was derived for each policy.

Table 7.2 – Mapping of Security Objectives to Security Policies

Policies	Objectives enforcing Policies	Rationale
P.ACCESS_BANNER *The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.*	O.DISPLAY_BANNER *The TOE will display an advisory warning regarding use of the TOE.*	O.DISPLAY_BANNER satisfies this policy by ensuring that the TOE displays a banner that provides authorized users with an advisory warning about the unauthorized use of the TOE.
P.ACCOUNTABILITY *The users of the TOE shall be held accountable for their actions within the TOE.*	O.AUDIT_GENERATION *The TOE will provide administrators with the necessary information for secure management of the TOE.* O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	Enforcement of this policy requires that users be uniquely identified [O.USER_IDENTIFICATION] and that their security relevant actions be monitored and recorded [O.AUDIT_GENERATION]. The recorded audit information can be selectively reviewed in search of any potential security violations [O.AUDIT_REVIEW].
P.AUTHORIZATION *The TOE shall limit the extent of each user's abilities in accordance with the TSP.*	O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.* O.USER_IDENTIFICATION *The TOE will uniquely identify users*	O.ACCESS supports this policy by requiring the TOE to uniquely identify authorized users [O.USER_IDENTIFICATION] prior to allowing any TOE access or any TOE mediated access on behalf of those users. Within the TOE, O.PROTECT provides mechanisms to prevent user data from unauthorized disclosure and modification.

P.AUTHORIZED_USERS *Only those users who have been authorized to access the information within the TOE may access the TOE.*	O. ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.*	Access control policies are used to define the access permitted to the system and its resources. These policies are supported by the implementation of authorized user attributes that identify the user-allowed accesses to TOE information. O.ACCESS supports this policy by ensuring that users only gain authorized access to TOE information and its resources by checking user attributes before system use.
P.I_AND_A *All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.*	O.USER_AUTHENTICATION *The TOE will verify the claimed identity of users.* O.USER_IDENTIFICATION *The TOE will uniquely identify users.*	In support of the policy to identify and authenticate a user before access is granted to any controlled resources, O.USER_IDENTIFICATION and O.USER_AUTHENTICATION will uniquely identify and authenticate the claimed authorized users.
P.INDEPENDENT_TESTING *The TOE must undergo independent penetration testing and some vulnerability analysis.*	O.FUNCTIONAL_TESTING *The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements.* O.PENETRATION_TEST *The TOE will undergo independent penetration testing to demonstrate that the design and implementation of the TOE prevents users from violating the TOE's security policies.* O.VULNERABILITY_ANALYSIS *The TOE will undergo appropriate vulnerability analysis for vulnerabilities that are obvious.*	This policy requires the TOE to undergo independent testing to verify its reliability and security. O.FUNCTIONAL_TESTING demonstrates the TSF satisfies the appropriate security functional requirements. O.PENETRATION_TESTING requires the TOE to undergo penetration testing and demonstrate that the design and implementation of the TOE do not allow users to violate the TOE's security policies. O.VULNERABILITY_ANALYSIS requires the TOE to undergo appropriate vulnerability analysis for vulnerabilities that are obvious.

P.NEED_TO_KNOW *The TOE must limit the access to information in protected resources to those authorized users who have a need to know that information.*	O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.* O.DISCRETIONARY_ACCESS *The TOE will control accesses to resources based upon the identity of users and groups of users.* O.DISCRETIONARY_USER_CONTROL *The TOE will allow authorized users to specify which resources may be accessed by which users and groups of users.* O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	The need-to-know policy is satisfied by the discretionary access control rules. O.DISCRETIONARY_ACCESS protects resources based on the identity of authorized users where the access to objects is directed by owners of the object [O.DISCRETIONARY_USER_CONTROL]. O.PROTECT enforces these policy rules by providing the mechanisms to protect the user data from disclosure and modifications and lastly, O.ACCESS ensures that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
P.RATINGS_MAINTENANCE *Procedures to maintain the TOE's rating must be in place to maintain the TOE's rating once it is evaluated.*	O.RATINGS_MAINTENANCE *Procedures to maintain the TOE's rating will be documented.*	O.RATINGS_MAINTENANCE satisfies this policy by ensuring that the TOE developer has procedures and mechanisms in place to maintain the evaluated rating that is ultimately awarded the TOE. The developer must provide a plan that identifies the certified version of the TOE and its life cycle process. Identifies any plans for new releases of the TOE to include a description of the changes included in the new release and a security impact analysis of implementing the new changes. Assign and identify the TOE's developer security analyst and ensure that they follow documented procedures. TOE components must be categorized by security relevance. The categorization scheme must be documented and followed for changes to the TOE.
P.ROLES *The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.*	O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	To appropriately administer the system, O.ADMIN_ROLE requires the system to provide multiple administrator roles to isolate actions performed by these different roles. To completely satisfy this policy, separate roles must be assigned separate individuals.

P.SYSTEM_INTEGRITY *The TOE shall provide the ability to periodically validate its correct operation and, with the help of administrators, it must be able to recover from any errors that are detected.*	O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.* O.CORRECT_TSF_OPERATION *The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.*	In order for an organization to place a measure of trust in the security features of a TOE, the TOE must include mechanisms that provide some measure of confidence in its correct functioning during its operational life-cycle. To provide such confidences, O.TRUSTED_SYSTEM_OPERATION provides self-tests that run during system start up, or at the request of the system administrator, and ensure that the TOE security mechanisms are operating properly When a security failure occurs and the TOE self-tests determine that the TOE is not operating in accordance with its security policies, O.RECOVERY provides the mechanisms that will return the TOE to a known secure operating state such that the security policies are enforced on all future processing.
P.TRACE *The TOE shall provide the ability to review the actions of individual users.*	O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.*	A common organizational security policy is to maintain records allowing for individuals to be held responsible for the actions that they take with respect to organizational assets. Information can be one of the most valuable assets that an organization possesses. To satisfy this policy, O.AUDIT_REVIEW provides suitable mechanisms to accurately and selectively review those records by authorized personnel to provide accountability at the individual user level to determine any potential security violation.
P.TRUSTED_RECOVERY *Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.*	O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.*	After a failure or other discontinuity, the security condition of the TOE may be unknown. O.RECOVERY provides procedures and/or mechanisms to ensure that recovery to a known secure state is obtained without a protection compromise.

P.VULNERABILITY ANALYSIS *The TOE will undergo some vulnerability analysis to demonstrate the design and implementation of the TOE does not contain any obvious flaws.*	O.VULNERABILITY_ANALYSIS *The TOE will undergo appropriate vulnerability analysis for vulnerabilities that are obvious.*	O.VULNERABILITY_ANALYSIS satisfies this policy by ensuring that an independent analysis is performed on the TOE and penetration testing based on that analysis is performed. Having an independent party perform the analysis helps ensure objectivity and eliminates preconceived notions of the TOE's design and implementation that may otherwise affect the thoroughness of the analysis. The level of analysis and testing requires that an attacker with a moderate attack potential cannot compromise the TOE's ability to enforce its security policies.

7.3 Objectives derived from Assumptions

78 Each of the identified security assumptions implies a set of security objectives to be met. The table below summarizes this mapping; this is then followed by explanatory text of how this mapping was derived for each assumption.

Table 7.3 – Mapping of Security Objectives to Assumptions

Assumptions	Objectives enforcing Assumptions	Rationale
A.PHYSICAL *It is assumed that appropriate physical security is provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.*	OE.PHYSICAL *Physical security will be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information.*	Physical security must be provided within the domain for the value of the IT assets protected by the TOE and the value of the stored, processed, and transmitted information. [OE.PHYSICAL].

7.4 Requirements Rationale

79 Each of the security objectives identified in sections 7.1 and 7.2 are addressed by one or more security requirements. Table 7.4 below provides the mapping from security requirements to security objectives, as well as a rationale that discusses how the security objective is met. Definitions are provided (*in italics*) below each security objective so the PP reader can reference these without having to go back to section 4.

Table 7.4 – Mapping of Security Requirements to Objectives

Objectives from Policies/Threats	Requirements Meeting Objectives	Rationale
O.ACCESS *The TOE will ensure that users gain only authorized access to it and to resources that it controls.*	FDP_ACC.1 FDP_ACF.1 FIA_AFL.1 FIA_UAU.1 FIA_UID.1 FMT_MOF.1(1-2) FMT_MSA.1 FMT_MTD.1(1-7)	The TOE must protect itself and the resources it controls from unauthorized access. FDP_ACF.1 specifies the DAC policy rules that will be enforced by the TSF and determines if an operation among subjects and named objects is allowed. Furthermore, it specifies the rules to explicitly authorize or deny access to a named object based upon security attributes. FIA_AFL.1 provides a detection mechanism for unsuccessful authentication attempts. The requirement enables an authorized administrator configurable threshold that prevents unauthorized users from gaining access to authorized user's account by guessing authentication data. This mechanism prevents access by either disabling the targeted account. Thus, limiting an unauthorized user's ability to gain unauthorized access

	FMT_REV.1(1) FMT_REV.1(2) FMT_SAE.1 FPT_RVM.1 FTA_LSA.1 FTA_SSL.1 FTA_SSL.2 FTA_TAB.1 FTA_TSE.1	to the TOE. FMT_REV.1 (1) ensures that the authorized administrator has the ability to revoke security attributes to a specific user. This revocation is immediate and helps authorized administrators control the ability of authorized users to log in or perform privileged operations. FMT_REV.1 (2) ensures that the authorized administrator and owners of named objects have the ability to revoke security attributes to a specific user. This revocation occurs when an access check is made and helps authorized administrators and owners control the ability of users accessing named objects. FPT_RVM.1 ensures that the TSF makes policy decisions on all access attempts to the TOE resources. Without this non-bypassability requirement, the TSF could not be relied upon to completely enforce the security policies. While untrusted users are only intended to access public objects, this requirement ensures they cannot access other objects provided by the TOE. This requirement also ensures that an administrator acting in a role only has access to the functions designated for that role. FTA_LSA.1 ensures that the scope of roles and user privileges are restricted based on location, time, and day. The intent of this requirement is to allow or disallow the assumption of roles or the effectiveness of user privileges based on the location where the session was established or the date/time of session establishment. “Location” refers to what ever means the TOE uses to identify a point of entry for interactive user session establishment. The adequacy of this means is determined by other requirements (e.g., FPT_SEP, AVA_VLA). FTA_SSL.1 is used to prevent unauthorized access to the TOE and its resources when an interactive session is left unattended. This requirement ensures that the interactive session will lock by making the visible contents unreadable after a specified time interval of session inactivity. The authorized user needs to re-authenticate to unlock his session. FTA_SSL.2 is used to ensure that unauthorized access to the TOE and its resources when an interactive session is left unattended. It enables the authorized user to lock his interactive session before leaving the session unattended. This eliminates any chance for any user to acquire unauthorized access to an unattended session because there is no time interval of inactivity before the

		session is locked. The authorized user needs to re-authenticate to unlock his session. FTA_TSE.1 is used to control the ability of an authorized user to establish a TOE session. The ability of a the administrator to determine which users are able to establish a session at a specific range of time, and from a specific location affords the TOE the ability to limit the exposure of the TOE to an attacker attempting to establish a session. For example, if the authorized user John Doe is only allowed to establish a session from 8 to 5, Monday through Friday, anyone attempting to establish a session as John Doe other than during those hours would not succeed, regardless of possession of John Doe's authentication data.
O.ACCESS_HISTORY *The TOE will display information (to authorized users) related to previous attempts to establish a session.*	FTA_TAH.1	FTA_TAH.1 is used to provide information about previous interactive sessions (i.e., date, time, and location). This information is displayed to the authorized user upon each successful interactive session establishment. This requirement gives the authorized users the ability to verify their last successful interactive session and thus, is a means for determining if the previous successful interactive session establishment was authorized or not.
O.ADMIN_ROLE *The TOE will provide administrator roles to isolate administrative actions.*	FMT_SMR.1	FMT_SMR.1 is used to maintain the role of authorized administrator and ensures the TSF shall be able to associate authorized users with roles.
O.ADMIN_GUIDANCE *The TOE will provide administrators with the necessary information for secure management.*	ADO_IGS.1 AGD_ADM.1	ADO_IGS.1 provides the procedures necessary for the secure installation, generation, and start-up of the TOE. AGD_ADM.1 provides administrative guidance to configure and administer the TOE securely for the IT environment it is intended to operate. The guidance also provides information about the corrective measures necessary when a failure occurs (i.e., how to bring the TOE back into a secure state).
O.AUDIT_GENERATION *The TOE will provide the capability to detect and create records of security relevant events associated with users.*	FAU_GEN.1 FAU_SEL.1 FIA_USB.1 FMT_MOF.1(1) FMT_MOF.1(2) FPT_STM.1	FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the authorized administrator has the ability to audit any security relevant event that takes place in the TOE. This requirement also defines the information that must be contained in the audit record for each auditable event. There is a minimum of information that much is present in every audit record and this requirement defines that, as well as the additional information that must be recorded for each auditable event. This requirement also places a requirement on the level of detail that is recorded on any additional security

	ADV_FSP.1 ADV_HLD.1 ADV_SPM.1	functional requirements an ST author adds to this PP. FAU_SEL.1 allows the authorized administrator to configure which auditable events will be recorded in the audit trail. This provides the administrator with the flexibility in recording only those events that are deemed necessary by site policy, thus reducing the amount of resources consumed by the audit mechanism. FIA_USB.1 plays a role is satisfying this objective by requiring a binding of security attributes associated with users that are authenticated with the subjects that represent them in the TOE. This only applies to authenticated users, since the identity of unauthenticated users cannot be confirmed. Therefore, the audit trail may not always have the proper identity of the user that causes an audit record to be generated (e.g., an attacker/user providing another user's user identifier). FPT_STM.1 ensures that the time stamps used to create the audit records are reliable. The time and date included in the time stamp is crucial when generating the audit information to ensure accountability.
O.AUDIT_PROTECTION *The TOE will provide the capability to protect audit information.*	FAU_SAR.2 FAU_STG.1 FMT_MTD.1(1-7) ADV_SPM.1	The audit trail must be protected so that only authorized users and authorized administrators may access it or delete it. FAU_SAR.2 ensures that only authorized users have read access to audit information and FAU_STG.1 ensures that audit information is not modified and protects it from unauthorized deletions. FMT_MTD.1(3) provides protection of audit information.
O.AUDIT_REVIEW *The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.*	FAU_SAR.1 FPT_STM.1 ADV_FSP.1 ADV_HLD.1 ADV_SPM.1	FAU_SAR.1 provides the ability for an authorized administrator to efficiently review audit records. This requirement also mandates the audit information be presented in a manner that is suitable for the administrators to interpret the audit trail.
O.CHANGE_MANAGEMENT *The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development life-cycle.*	ACM_CAP.3 ACM_SCP.2 ALC_FLR.2	ACM_SCP.2 is necessary to define what items must be under the control of the CM system. This requirement ensures that the TOE implementation representation, design documentation, test documentation (including the executable test suite), user and administrator guidance, CM documentation and security flaws are tracked by the CM system. ALC_FLR.2 plays a role in satisfying the "analyzed"

		portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws.
O.CORRECT_TSF _OPERATION *The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.*	FMT_MSA.2 FPT_AMT.1	This objective requires one-security functional requirements in the FPT class to be met: FPT_AMT. This functional requirement provides the end user with the capability to ensure the TOE's security mechanism continues to operate correctly in the field. FPT_AMT.1 has been refined to ensure end user tests exist to demonstrate the correct operation of the security mechanisms required by the TOE that are provided by the hardware. Hardware failures could render a TOE's software ineffective in enforcing its security policies and this requirement provides the end user the ability to discover any failures in the hardware security mechanisms. Additionally, O.CORRECT_TSF_OPERATION requires FMT_MSA.2. This requirement ensures that only valid values are accepted for security attributes. The values that are accepted as valid for a specific security attribute must fall within the appropriate range for that attribute (e.g., the password length attribute must be a non-negative integer).
O.DISCRETIONARY_ACCESS *The TOE will control accesses to resources based upon the identity of users and groups of users.*	FDP_ACC.1, FDP_ACF.1, FDP_ITT.1, FIA_USB.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(3), FMT_MTD.1(4), FMT_MTD.1(5), FMT_MTD.1(6), FMT_MTD.1(7), FMT_REV.1(1), FMT_REV.1(2), FPT_RVM.1 ADV_FSP.1, ADV_HLD.1, ADV_SPM.1	FDP_ACF.1 defines the Discretionary Access Control rules to determine if any operation between subjects and named objects is allowed. These rules are based on the identity of the users and their group memberships. FIA_USB.1 defines the associations between user security attributes and subjects acting on behalf of that user by which policy decisions are based upon. FMT_MSA.3 ensures that the TOE provides protection by default for all named objects at creation time. This may allow authorized users to explicitly specify the desired access controls upon the object at its creation, provided that there is no window of vulnerability through which unauthorized access may be gained to newly-created objects. FPT_RVM.1 ensures that the Discretionary Access Control policy is not bypassed. The discretionary aspect of the policy is that users who control access to objects can set that access to be restrictive or permissive to other users at their discretion. The policy is to be

		always enforced, never optional. ADV_SPM.1 requires the developer to provide an informal model of the Discretionary Access Control policy. Modeling the policy helps understand and reduce the unintended side-effects that occur during the TOE's operation that might adversely affect the TOE's The discretionary access control mechanism is described in terms of its purpose ADV_FSP.1 and its external interfaces (ADV_HLD.1). The discretionary access control policy is defined by ADV_SPM.1.
O.DISCRETIONARY_USER_CONTROL *The TOE will allow authorized users to specify which users and groups of users may access which resources.*	FDP_ACF.1	FDP_ACF.1 allows administrators and object owners to change the object's attributes used for the enforcement of the discretionary access control policy.
O.DISPLAY_BANNER *The TOE will display an advisory warning regarding use of the TOE.*	FIA_UAU.1, FIA_UID.1, FTA_TAB.1	Before identification and authentication and the establishment of a user session, the TOE allows limited access by any potential users of the system in order to convey warnings and agreements for system use. Through this limited access before establishing a user session, the TSF displays an authorized, administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE [FTA_TAB.1]. In typical applications a user who continues session establishment procedures (including their successful identification and authentication) after display of the notice and warning banner effectively acknowledges the banner content and consents to the stated conditions. This banner of information can be critical in supporting legal actions related to the use of the TOE.
O.FUNCTIONAL_TESTING *The TOE will undergo appropriate security functional testing, that demonstrates the TSF satisfies the security functional requirements.*	ATE_COV.2 ATE_DPT.1 ATE_FUN.1 ATE_IND.2	In order to satisfy O.FUNTIONAL_TESTING, the ATE class of requirements is necessary. Requirements fall into two categories; those that are levied on the developer to create and document the security test suite and those that are levied on the evaluation team to independently verify the testing results. The first category comprises ATE_FUN.1, ATE_COV.2 and ATE_DPT.1. The component ATE_FUN.1 requires the developer to provide the necessary test documentation to allow for an independent analysis of the developer's security functional test coverage. ATE_DPT.2 requires the developer to provide a test coverage analysis that demonstrates depth of coverage of the test suite. The second category comprises ATE_IND.2 which requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. This component also

		requires an independent party to attempt to craft functional tests that address functional behavior that is not demonstrated in the developer's test suite. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated.
O.INSTALL_GUIDANCE *The TOE will be delivered with the appropriate installation guidance to establish and maintain TOE security.*	ADO_DEL.1, ADO_IGS.1	Once secure delivery from the developer to the user has occurred, appropriate installation guidance should be used for the secure installation, generation and start-up of the TOE at the user's site. This phase securely transitions the TOE from the developer's configuration control to the user's operational environment. ADO_IGS.1 requires the developer to describe and document the procedures needed for secure TOE installation, generation, and start-up. ADO_IGS.1 also requires an evaluator to confirm that the procedures meet all the requirements for content and presentation of evidence, and that the procedures result in a secure configuration for the TOE.
O.MANAGE *The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.*	FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.3, FAU_STG.3, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(3), FMT_MTD.1(4), FMT_MTD.1(5), FMT_MTD.1(6), FMT_MTD.1(7), FMT_SAE.1, FTA_LSA.1 ADO_DEL.1, ADO_IGS.1, AGD_ADM.1	In a variety of ways the TOE supports authorized administrators in the management of security functions, security attributes and data while also restricting unauthorized use. For example, the TOE provides for and restricts the following actions to authorized administrators only (except where specifically noted): • Disable and enable the audit functions, and specify which events are audited [FMT_MOF.1 (1)] • Create, initialize, change default, modify, delete, clear, append, query, etc. the values of security attributes associated with user authentication data [FMT_MOT.1 (2)]. • Change the value of object security attributes. (Object owner is also allowed to perform this action.) [FMT_MSA.1]. • Provide restrictive default values for security attributes, and specify alternative initial values to override the default values when an object or information is created. [FMT_MSA.3]. • Create, initialize, change default, modify, delete, clear, append, query, etc. the security-relevant TSF data (except audit records, user security attributes, authentication data, and critical security parameters) [FMT_MTD.1 (1)]. • Query, delete, and clear audit records [FMT_MTD.1 (2)]. • Initialize user security attributes. [FMT_MTD.1 (4)]. • Modify user security attributes, other than authentication data. [FMT_MTD.1 (5)].

		• Modify authentication data. (Also allows users authorized to modify their own authentication data to do so.) • Specify an expiration time for authorized user authentication data. [FMT_SAE.1].
O.PENETRATION_TEST *The TOE will undergo independent penetration testing to demonstrate that the design and implementation of the TOE prevents users from violating the TOE's security policies.*	AVA_VLA.1	These analyses for vulnerabilities must performed by the developer to identify the presence of obvious security vulnerabilities, and show that the TOE cannot be exploited in its intended environment [AVA_VLA.1].
O.PROTECT *The TOE will provide mechanisms to protect user data and resources.*	FDP_ACF.1, FDP_ITT.1, FDP_RIP.2, FIA_SOS.1, FIA_UAU.1, FIA_UAU.6, FIA_UAU.7, FIA_UID.1, FMT_MSA.1, FMT_REV.1(1), FMT_REV.1(2), FPT_RVM.1, FPT_SEP.1, FTA_SSL.1, FTA_SSL.2	O.PROTECT requires mechanisms be provided by the TOE to protect FIA_SOS.1 prescribes the metrics that must be satisfied for user authentication. If a user can't authenticate, he or she will not have the ability to access user data and resources. FIA_SOS.2 requires that the authentication mechanisms provide the ability for authorized users to have a "secret" in a manner that cannot be guessed at random in less than one in 5 x 1015. FIA_UAU.7 ensures that no feedback that affects the ability of users to circumvent the authentications mechanism is presented during the authentication process. The TOE is allowed to provide information that would allow the user to use the authentication mechanism in a correct manner (e.g., press CTRL-ALT-DELETE, slide card quickly, center your finger and press firmly, speak louder and slowly), but not provide information that may allow alteration to their presentation that would thwart the mechanism. FPT_RVM.1 requires the TSF enforce a policy before each user action to protect resource in question. To protect user data and resources, FDP_ACF.1 and FMT_REV.1(2) require a Discretionary Access policy and rules that ensures that correct access to named objects by subjects acting on behalf of users. In addition, FDP_ITT.1 prevents and rules that ensures the correct access to named objects by subjecting acting on behalf of users. In addition, FDP_ITT.1 prevents the disclosure and modification of user data while being transmitted between physically separate parts of the TOE. To ensure that user data is not disclosed before a resource is reused, FDP_RIP.2 ensures that the user data contained within the object is not available to another user thus protecting the user data.
O.RATING_MAINTENANCE *Procedures to maintain the TOE's rating will be documented.*		

O.RECOVERY *Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as from system failure or discontinuity.*	FPT_RCV.1, FPT_STM.1, FPT_TRC_EXP.1	FPT_RCV.1 ensures that the system enters a maintenance mode allowing the system to be returned to a secure state after a failure or service discontinuity. In a secure state, all security policies are enforced; in addition, the critical areas of the cryptography are zeroized, are ready to be reloaded, and are inaccessible to processes.
O. REPLAY_DETECTION *The TOE will provide a means to detect and reject the replay of authentication data, as well as TSF data and security attributes.*	FPT_ITT.3	O.REPLAY_DETECTION is satisfied by the requirement FPT_ITT.3. This requirement ensures the TOE detects attempted modification, insertion and replay of TSF data.
O.RESIDUAL_INFORMATION *The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.*	FDP_RIP.2, FPT_RCV.1, FTA_SSL.1, FTA_SSL.2	For all other resources, FDP_RIP.2 ensures that contents of resources are unavailable to subjects other than those explicitly granted access to the data.
O.RESOURCE_SHARING *The TOE shall provide mechanisms that mitigate user attempts to exhaust TOE resources (i.e., system memory, persistent storage, and processing time).*	FRU_RSA.1(1) FRU_RSA.1(2) FRU_RSA.1(3) FTA_MCS.1	This objective requires mechanisms to prevent authorized users (or software unknowingly acting on their behalf) from exhausting important resources controlled by the TOE in a manner that adversely impacts other users or programs. The TOE is required to enforce a limit on the amount of resources a given authorized user may successfully be granted. The resources that are controlled are: CPU time, disk space, system memory, and user accounts. FRU_RSA.1 (iterations 1, 2, and 3) is intended to enforce the notion that a single authorized user may only be allocated a "preset maximum" amount of resource. The iterations cover the major resources that are required to offer confidence that entities executing on the TOE are not "starved for resources" and will be allowed to initiate and complete execution. FTA_MSA.1 identifies user accounts as a system resource that could be exhausted (through multiple concurrent "logons" of a single individual). The requirement mandates that the administrator be able to limit the number of concurrent logon sessions by a single user. This ensures that a single individual could not mount a denial-of-service attack using multiple sessions as launching points. Resources (e.g., memory contained on the network card) that are not covered by the above are subject to denial of service attacks. Denial-of-service attacks of these resources should be addressed via other mechanisms such as redundant hardware.

O.REFERENCE_MONITOR *The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure and ensures that the security policies implemented by the TOE are always invoked.*	FPT_ITT.3 FPT_RCV.1 FPT_TRC_EXP.1	This objective requires the protection of the TSF (and its data) from external interference, tampering or inappropriate disclosure by mandating that the TSF create and maintain a domain for its execution. Domain is defined as the logical area that the TSF provides for itself in which to operate. Common mechanisms include hardware execution domains (e.g., processor execution rings as well as other isolation mechanisms that protect TSF data when it is in transit to other TSF components.) The requirements that implement this objective fall into two categories. The first category mandates mechanisms to implement a secure domain for execution. The second category mandates that if the TSF (for some reason) moves into an unknown or unconnected state, that it has a way to recover to a known or connected state. This ensures that the TSF can continue to protect itself even after unexpected interruptions. Requirements included in the first category are FPT_SEP.2, and FPT_ITT.3. . FPT_ITT.3 was chosen to protect TSF data in transmission between remote portions of the TSF and also requires that mechanisms be in place to protect against man-in-the-middle replay attacks which could attempt to interfere with the TSF policy being enforced. Requirements included in the second category are FPT_RCV.1 and FPT_TRP_EXP.1. FPT_RCV.1 is used to ensure that the TSF offers a mechanism to recover from a failed state by mandating that the TSF provide maintenance mode from which to re-initiate (or establish) a known (secure) state. This ensures that once the TSF has established a domain for its own execution it can always return to that state with confidence that this domain continues to be present. FPT_TRP_EXP.1 is used to address distributed TSFs and the fact that portions of these TSF may become disconnected over time. A disconnected portion of the TSF does not always suggest an insecure state or discontinuity of service (referenced in FPT_RCV.1). Instead, this requirement addresses the situation when a portion of a distributed TSF is disconnected from the rest of the TSF (with both pieces continuing service). Specifically, it requires that there be mechanisms provided by the TSF to ensure that upon reconnection, the TSF portions will become in sync over a reasonable time period.
O.SECURE_STATE *The TOE will be able to verify the integrity*	FPT_RCV.1	FPT_RCV.1 ensures that the TOE does not continue to operate in an insecure state when a hardware or software failure occurs. Upon the failure of the TSF

of the TSF code and cryptographic data.		software failure occurs. Upon the failure of the TSF self-tests (including the hardware tests required by FPT_AMT) the TOE will enter a mode where it can no longer be assured of enforcing its security policies. Therefore, the TOE enters a state that disallows traffic flow and requires an administrator to follow documented procedures that instruct them on to return the TOE to a secure state. These procedures may include running diagnostics of the hardware, or utilities that may correct any integrity problems found with the TSF data or code. Solely specifying that the administrator reload and install the TOE software from scratch, while might be required in some cases, does not meet the intent of this requirement.
O.SOUND_DESIGN *The TOE will be designed using sound design principles and techniques. The TOE design, design principles and design techniques will be adequately and accurately documented.*	ALC_FLR.2, AVA_MSU.1, AVA_SOF.1, AVA_VLA.1, ADV_FSP.1, ADV_HLD.1, ADV_RCR.1, ADV_SPM.1	ADV_SPM.1 requires the developer to provide an informal model of the security policies of the TOE. Modeling these policies helps understand and reduce the unintended side-effects that occur during the TOE's operation that might adversely affect the TOE's ability to enforce its security policies. The ADV_RCR.1 is used to ensure that the levels of decomposition of the TOE's design are consistent with one another. This is important, since design decisions that are analyzed and made at one level (e.g., functional specification) that are not correctly designed at a lower level may lead to a design flaw. This requirement helps in the design analysis to ensure design decisions are realized at all levels of the design. The AVA_SOF.1 requirement is applied to the user authentication mechanism. For this TOE, the strength of function specified is medium. This requirement ensures the developer has performed an analysis of the authentication mechanism to ensure the probability of guessing a user's authentication data would require a high-attack potential, as defined in Annex B of the CEM.
O.SOUND_IMPLEMENTATIO N	ALC_FLR.2, ATE_COV.2, ATE_DPT.1, ATE_FUN.1, ATE_IND.2, AVA_MSU.1, AVA_SOF.1, AVA_VLA.1, ADV_FSP.1, ADV_HLD.1, ADV_IMP.1, ADV_RCR.1	ALC_FLR.2 plays a role in satisfying the "accurate instantiation" portion of this objective by requiring the developer to have procedures that address flaws that have been discovered in the product, either through developer actions (e.g., developer testing) or those discovered by others. The flaw remediation process used by the developer corrects any discovered flaws and performs an analysis to ensure new flaws are not created while fixing the discovered flaws. ATE_IND.2 requires an independent confirmation of the developer's test results, by mandating a subset of the test suite be run by an independent party. This component also requires an independent party to

		attempt to craft functional test that address functional behavior that is not demonstrated in the developer's test suite. Upon successful adherence to these requirements, the TOE's conformance to the specified security functional requirements will have been demonstrated
O.TRAINED_USERS *The TOE will provide authorized users with the necessary guidance for secure use of the TOE, to include secure sharing of user data.*	AGD_USR.1	O.TRAINED_USERS requires that user's procedures for the secure use of the TOE be documented. AGD_USR.1 states that the developer shall provide user guidance describing the functions and interfaces available to the non-administrative users of the TOE. The user guidance shall also describe the use of user-accessible security functions, and shall clearly present all user responsibilities necessary for secure operation of the TOE.
O.TRUSTED_SYSTEM_OPERATION	FIA_AFL.1, FIA_UAU.6, FIA_UAU.7, FIA_UID.1, FMT_SAE.1, FPT_AMT.1, FPT_RCV.1, FPT_STM.1, FPT_TRC_EXP.1, FTA_TAH.1 ADO_DEL.1, ADO_IGS.1, AGD_ADM.1	
O.USER_AUTHENTICATION *Users must authenticate their claimed identities (see O.USER_IDENTIFICATION) before they are allowed access to the TOE.*	FIA_SOS.1, FIA_UAU.1, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MSA.2, FMT_MTD.1(1), FMT_MTD.1(2), FMT_MTD.1(3), FMT_MTD.1(4), FMT_MTD.1(5), FMT_MTD.1(6), FMT_SAE.1, FTA_SSL.1, FTA_SSL.2 ADV_FSP.1, ADV_HLD.1, ADV_SPM.1	FIA_UAU.1 plays a role in satisfying this objective by ensuring that every user is authenticated before the TOE performs any TSF-mediated actions on behalf of that user. To verify the claimed identity of an authorized user, FIA_SOS.1 prescribes the metrics that must be satisfied. It provides the mechanism that will verify the secret for user authentication. The PP authors intentionally did not dictate that a password mechanism be required and allowed for other types of authentication mechanisms (e.g., a PIN, Token). In any case, FIA_SOS.1 requires that the authentication mechanism provide the ability for authorized users to have a "secret" in a manner that cannot be guessed at random in less than one in 5 x 1015. FTA_SSL.1 and FTA_SSL.2 ensure that the authorized user authenticates him or herself before accessing a locked interactive session. This eliminates any chance for any user to acquire unauthorized access to an unattended session. Active interactive sessions may be locked by a user or after a specified time interval of user

		inactivity configured by an authorized administrator.
O.USER _IDENTIFICATION *The TOE will uniquely identify users.*	FIA_ATD.1, FIA_UID.1, FIA_USB.1, FMT_SAE.1, FMT_SMR.1, ADV_FSP.1, ADV_HLD.1, ADV_SPM.1	FIA_UID.1 plays a role in satisfying this objective by ensuring that every user is identified before the TOE performs any TSF-mediated actions on behalf of that user. It also allows for the specification of a list of public objects that users are allowed read access before the user is identified.
O.VULNERABILITY _ANALYSIS The TOE will undergo appropriate vulnerability analysis for vulnerabilities that are obvious.	AVA_MSU.1, AVA_SOF.1, AVA_VLA.1	AVA_SOF.1 ensures that an analysis of the strength of the functions is performed. Even if a TOE security function cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat it because there is a vulnerability in the concept of its underlying security mechanisms. For those functions a qualification of their security

7.5 Explicit Requirements Rationale

81 The following explicit requirements have been included in this Protection Profile because the Common Criteria requirements were found to be insufficient as stated. The rationales for the explicit functional requirements included in this PP are explained in Table 7.5.

Table 7.5 – Rationale for Explicit Functional Requirements

Explicit Component	Rationale
FPT_TRC_EXP.1	FPT_TRC_EXP has been created to require timely consistency of replicated TSF data. Although there is a Common Criteria Requirement that attempts to address this functionality, if falls short of the needs of the environment in this protection profile. Specifically, FPT_TRC.1.1 states that "The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE." In the widely distributed environment of this PP's TOE, this is an infeasible requirement. For TOEs with a very large number of components, 100 percent TSF data consistency is not achievable and is not expected. Another concern lies in FPT_TRC.1.2 which states that when replicated parts of the TSF are "disconnected", the TSF shall ensure consistency of the TSF replicated data upon "reconnection". Upon first inspection, this seems reasonable, however, when applying this requirement it becomes clear that it dictates specific mechanisms to determine when a component is "disconnected" from the rest of the TSF and when it is "reconnected". This is problematic in this PP's environment in that it is not the intent of the authors to dictate that distributed TSF components keep track of

	connected/disconnected components. In general, to meet the needs of this PPs, it is acceptable to simply require a mechanism that provides TSF data consistency in a timely manner after it is determined that it is inconsistent.

7.6 Rational for Strength of Function

82 The TOE minimum strength of function is SOF-basic. The evaluated TOE is intended to operate in environments processing administrative, private, and sensitive/proprietary information. The minimum strength of function was chosen to be consistent with FIA_SOS.1 by providing a probability of successful authentication for a random attempt of less than one in 5×10^{15}. This security function is in turn consistent with the security objectives described in section 7.4.

7.7 Rationale for Assurance Rating

83 This protection profile has been developed for commercial-off-the-shelf (COTS) general-purpose operating systems in networked environments. The intended environments may process administrative, private, and sensitive/proprietary information. The type of information processed by the environment establishes the need for the TOE to be evaluated at an Evaluated Assurance Level 2 Augmented (EAL2+).

8. References

[1] Common Criteria Implementation Board, Common Criteria for Information Technology Security Evaluation, CCIB-98-026, Version 2.1, August 1999

[2] Department of Defense Chief Information Officer, Guidance and Policy for Department of Defense Information Assurance Memorandum No. 6-8510 dated 16 June 2000

[3] National Security Agency, Protection Profile For Single-level Operating Systems In Environments Requiring Medium Robustness Version 1.22, 23 May 2001

[4] Department of Defense Standard, Department of Defense Trusted Computer System Evaluation Criteria (Orange Book), December 1985

[5] Trusted Product Evaluation Program (TPEP) Trusted Computer System Evaluation Criteria (TCSEC) Interpretations

Appendix A — Acronyms

CC Common Criteria for Information Technology Security Evaluation Version 2.1

COTS Commercial Off-The-Shelf

DAC Discretionary Access Control

DoD Department of Defense

EAL Evaluation Assurance Level

IA Information Assurance

IT Information Technology

OS Operating System

PP Protection Profile

SF Security Function

SFP Security Function Policy

SOF Strength of Function

ST Security Target

TOE Target of Evaluation

TSC TSF Scope of Control

TSF TOE Security Functions

TSFI TSF Interface

TSP TOE Security Policy